



**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Gisele Bernardes da Silva

**ANÁLISE DE SOLUÇÕES PARA BUSCA POR SIMILARIDADE  
(*MATCHING*) DE DADOS MUSICAIS**

Florianópolis

2018

Gisele Bernardes da Silva

**ANÁLISE DE SOLUÇÕES PARA BUSCA POR SIMILARIDADE  
(*MATCHING*) DE DADOS MUSICAIS**

Trabalho de Conclusão do Curso de Graduação em Sistemas de Informação do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina como requisito para a obtenção do Título de Bacharel em Sistemas de Informação.  
Orientador: Prof. Dr. Ronaldo dos Santos Mello

Florianópolis

2018

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silva, Gisele Bernardes da  
Análise de Soluções para Busca por Similaridade  
(Matching) de Dados Musicais / Gisele Bernardes da Silva ;  
orientador, Ronaldo dos Santos Mello, 2018.  
139 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro Tecnológico,  
Graduação em Sistema de Informação, Florianópolis, 2018.

Inclui referências.

1. Sistema de Informação. 2. Recuperação da informação  
musical. 3. Busca por similaridade. 4. Dados musicais. 5.  
Áudio digital. I. Mello, Ronaldo dos Santos. II.  
Universidade Federal de Santa Catarina. Graduação em  
Sistema de Informação. III. Título.

Gisele Bernardes da Silva

**ANÁLISE DE SOLUÇÕES PARA BUSCA POR SIMILARIDADE  
(*MATCHING*) DE DADOS MUSICAIS**

Este Trabalho de Conclusão do Curso de Graduação foi julgado adequado para a obtenção do Título de “Bacharel em Sistemas de Informação”, e aprovado em sua forma final pelo Curso de Sistemas de Informação.

Florianópolis, 23 de Novembro 2018.

---

Prof. Cristian Koliver, Dr.  
Coordenador do Curso

---

Prof. Dr. Ronaldo dos Santos Mello  
Orientador

**Banca Examinadora:**

---

Prof. Dr. Renato Fileto

---

Prof. Dr. Raul Sidnei Wazlawick



Aos meus pais, cuja fé em mim me ensinaram a ter fé em mim mesmo.

## AGRADECIMENTOS

Sou grata aos meus pais pela vida, pois sem vocês minha existência não seria possível. Sou grata pelo carinho, apoio e amor incondicional depositados em mim todos os dias, e desta forma, posso ser quem sou hoje. Por serem meu exemplo de esforço e dedicação, por me mostrarem que eu sou capaz de trilhar o caminho que desejo e conquistar o que almejo.

Sou grata ao meu namorado, por todo o apoio, paciência, e compreensão da minha ausência em horas de estudo.

Sou grata aos meus amigos, pelo companherismo, alegrias, tristeza e principalmente pela amizade. Foi a cumplicidade do dia-a-dia de vocês e o nosso esforço como equipe que me ajudaram a chegar aonde cheguei.

Sou grata à universidade pela oferta do curso e à oportunidade de fazer parte da família nesse período de aprendizado.

Sou grata aos professores, pelo tempo e esforço em compartilhar todo o seu conhecimento, em especial ao meu orientador, pelo empenho dedicado à elaboração deste trabalho.

Sou grata a todos que direta ou indiretamente fizeram parte da minha formação.

Por fim, sou grata à Deus, pela oportunidade de estar nesta Terra e de poder conviver com essas pessoas, sempre aprendendo e evoluindo.

*A música, está em tudo ao nosso redor. Tudo o que temos que fazer é ouvir.*

Sheridan (2008)

## RESUMO

O som não é algo que podemos ver com nossos olhos. Então, o que é som? O som é a variação da pressão do ar. Sendo assim, a forma de produzir um determinado som depende da maneira como a pressão do ar varia. Representar o som numericamente é chamado de *digitalização*, ou seja, é representar uma onda sonora (áudio analógico) em código binário (áudio digital). Um processo de transformação e compressão do som, reduzindo o seu tamanho, pois armazenar dados de áudio em formato analógico consome muito espaço. Em paralelo, tornou-se possível codificar arquivos de áudio com mais informação do que apenas o nome do arquivo, utilizando os *metadados*. Com os metadados é possível nomear, descrever, catalogar e indicar os direitos de autor de um arquivo de áudio digital. Desta forma, um arquivo de áudio digital é composto por metadados e som digitalizado, sendo assim, um dado musical. O dado é a representação física de um evento no tempo e espaço que não agrega fundamento, não podendo ser possível entender o que ele representa ou para que ele existe, porém, ao incluir um “significado” no dado e gerar sentido para quem o ouve e ficando claro ou não a que se refere, é gerada a informação. Uma informação musical apresenta determinadas especificidades de comportamento na sua produção, objetivação e uso. Assim, a música tem diferentes significações para cada indivíduo. A música era um meio de comunicação exclusivamente presencial e com a evolução dos inventos tecnológicos, a música ultrapassa os limites físicos da mídia, mergulhando no universo digital. Desta forma, o problema de representação e o processo de construção de sistemas de processamento e recuperação musicais, agrava-se com a necessidade de desenvolvimento de sistemas com estruturas internas o mais compatível possível com as visões ou desejos dos usuários. Portanto, a relevância deste trabalho contribui diretamente, para agregar conhecimento com o estudo sobre a recuperação da informação musical, que auxiliarão no desenvolvimento futuro de soluções, para busca por similaridade de dados musicais. Especificamente, este trabalho visa apresentar e comparar soluções para recuperação de informação musical. A intenção é analisar soluções que não necessariamente buscam dados musicais apenas através do casamento direto de parâmetros de entrada para a busca, como título da música, palavras-chave ou um áudio com parte da música, mas também através do casamento aproximado (ou similar) destes parâmetros.

**Palavras-chave:** Recuperação da informação musical. Busca por similaridade. Dados musicais. Áudio digital. Metadados.

## ABSTRACT

Sound is something we can't see. So, what is sound? Sound is the variation of air pressure. The way to produce a certain sound depends the air pressure varies. Representing the sound numerically is called *scan*, it is to represent a sound wave (analog audio) in binary code (digital audio). A process of transformation and compression of sound, reducing its size, since storing audio data in analog format consumes a lot of space. In parallel, it became possible to encode audio files with more information than just the file name, using *metadata*. With metadata it is possible to name, describe, catalog and indicate the copyright of a digital audio file. In this way, a digital audio file is composed of metadata and digitized sound, thus being a musical data. The data is the physical representation of an event in time and space that does not add a foundation, and it can not be possible to understand what it represents or for what it exists, however, by including a "meaning" in the given and generating meaning for the listener and whether or not it is clear what it refers to, the information is generated. A musical information presents certain specificities of behavior in its production, objectification and use. Thus, music has different meanings for each individual. Music was a means of exclusively on-site communication and with the evolution of technological inventions, music surpasses the physical limits of the media, plunging into the digital universe. In this way, the problem of representation and the process of construction of musical processing and recovery systems is aggravated by the need to develop systems with internal structures as compatible as possible to the visions or desires of the users. Therefore, the relevance of this work contributes, directly, to aggregate knowledge with the study on the retrieval of musical data information that will aid in the future development of solutions for searching for similarity of musical data. Specifically, this work aims to present and compare solutions for music information retrieval. The intention is to analyze solutions that do not necessarily search for musical data only through direct marriage of input parameters to the search, such as song title, keywords or an audio with part of the song, but also through approximate (or similar) these parameters.

**Keywords:** Retrieval of musical information. Search for similarity. Musical data. Digital audio. Metadata.

## LISTA DE FIGURAS

Figura 1	Representação gráfica, no domínio temporal, de uma forma de onda senoidal	20
Figura 2	Formas de onda simples e seus timbres.....	20
Figura 3	Relação entre elementos do som e da música.....	22
Figura 4	Forma de onda analógica (curva preta) e representação digitalizada (as amostras digitalizadas são indicados pelas caixas cinzentas). Neste exemplo, a forma de onda é amostrada em 24 pontos e a quantização dos valores emprega um esquema de codificação de 4 bits (16 valores possíveis).....	23
Figura 5	Representação do modelo de recuperação de informação musical.....	36
Figura 6	Diagrama de identificação de áudio.....	37
Figura 7	Processo da Técnica HPS.....	38
Figura 8	Processo de extração de características.....	39
Figura 9	Modelagem da identificação de áudio.....	39
Figura 10	K-Nearest Neighbors.....	43
Figura 11	Usando o algoritmo K-means para encontrar três grupos nos dados de exemplo.....	45
Figura 12	Conjunto de objetos de dados em clusters.....	45
Figura 13	Um cluster hierárquico de quatro pontos mostrado como um dendograma e como grupos aninhados.....	47
Figura 14	Eps, pontos de centro, de limite e de ruído.....	48
Figura 15	Execução do algoritmo de DTW. (A) Séries C e Q. (B) Matriz de distâncias. (C) Alinhamento das duas séries com o caminho mínimo.....	50
Figura 16	Distância Euclidiana entre os pontos A e B.....	52
Figura 17	Distância de Manhattan (linhas roxa, verde e vermelha).....	53
Figura 18	Distância de Manhattan (linhas rosa, verde e vermelha) e Distância Euclidiana (linha azul).....	54

Figura 19 Longest Common Subsequence.....	55
Figura 20 Modelagem do Sistema de QBH .....	56
Figura 21 Sistema de QBH.....	57
Figura 22 Análise de SMS e diagramas de bloco de síntese.....	60
Figura 23 Mapa topográfico com pequenos pontos para as músicas.....	63
Figura 24 MusicID.....	64
Figura 25 Shazam.....	65
Figura 26 SoundHound.....	67
Figura 27 Deezer.....	68
Figura 28 Spotify .....	69
Figura 29 SoundCloud .....	70
Figura 30 Musixmatch .....	72
Figura 31 ACRCcloud.....	73
Figura 32 Musipedia .....	75
Figura 33 AMUSE.....	76
Figura 34 CLAM.....	77
Figura 35 Java MIR.....	79
Figura 36 MIRtoolbox .....	80
Figura 37 MusicMiner.....	82
Figura 38 Aplicativo Tunebot .....	83
Figura 39 Tunebot Web.....	84

## LISTA DE TABELAS

Tabela 1	Diferenças entre as versões ID3 .....	25
Tabela 2	Diferenças entre as formas de armazenamento de metadados .....	26
Tabela 3	Comunidades de MIR .....	31
Tabela 4	Análise das soluções comerciais: Critério de Eficiência .....	89
Tabela 5	Análise das soluções comerciais: Critério de Adequação Funcional.....	95
Tabela 6	Análise das soluções comerciais: Critério de Usabilidade .....	96
Tabela 7	Análise das soluções acadêmicas: Critério de Eficiência.....	96
Tabela 8	Análise das soluções acadêmicas: Critério de Adequação Funcional.....	97



## LISTA DE ABREVIATURAS E SIGLAS

UFSC	Universidade Federal de Santa Catarina .....	18
KB	<i>Kilobyte</i> .....	24
MB	<i>Megabyte</i> .....	24
BD	Banco de Dados .....	25
LP	<i>long-play</i> .....	28
DVD	<i>Digital Versatile Disc</i> .....	29
SGBD	Sistema de Gerenciamento de Banco de Dados .....	29
ISMIR	<i>International Society for Music Information Retrieval</i> .....	35
MIREX	<i>Music Information Retrieval Evaluation eXchange</i> .....	35
CRP	<i>Cross Recurrence Plots</i> .....	48
RP	<i>Recurrence Plot</i> .....	48
DTW	<i>Dynamic Time Warping</i> .....	49
SMS	Spectral Modeling Synthesis .....	60
ESOM	<i>Emergent Self-Organizing Maps</i> .....	61
IA	Inteligência Artificial .....	87

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	14
1.1 OBJETIVOS	16
1.1.1 Objetivo Geral	16
1.1.2 Objetivos Específicos	16
1.2 METODOLOGIA	17
1.3 ESTRUTURA DO TRABALHO	17
<b>2 FUNDAMENTAÇÃO TEÓRICA</b>	19
2.1 CONCEITOS BÁSICOS SOBRE SOM	19
2.2 REPRESENTAÇÃO DO ÁUDIO DIGITAL	21
2.3 INFORMAÇÃO MUSICAL	25
2.4 ARMAZENAMENTO DA INFORMAÇÃO MUSICAL	28
2.5 RECUPERAÇÃO DA INFORMAÇÃO MUSICAL	30
2.5.1 Métodos e Algoritmos para Recuperação da Informação Musical	35
2.5.1.1 Audio Fingerprint	36
2.5.1.2 Classificação	40
2.5.1.3 Clustering	43
2.5.1.4 Cross Recurrence Plot	48
2.5.1.5 Dynamic Time Warping	49
2.5.1.6 Indexação	51
2.5.1.7 Medidas de Similaridade	51
2.5.1.8 Query by Humming	55
2.5.1.9 Recuperação por Conteúdo	58
2.5.1.10 Spectral Modeling Synthesis	60
2.5.1.11 Visualização	61
<b>3 SOLUÇÕES EXISTENTES</b>	64
3.1 SOLUÇÕES COMERCIAIS	64
3.1.1 MusicID	64
3.1.2 Shazam	65
3.1.3 SoundHound	66
3.1.4 Deezer	67
3.1.5 Spotify	68
3.1.6 SoundCloud	70

3.1.7 Musixmatch .....	71
3.1.8 ACRCcloud .....	72
3.1.9 Musipedia .....	73
3.2 SOLUÇÕES ACADÊMICAS .....	74
3.2.1 AMUSE .....	74
3.2.2 CLAM .....	76
3.2.3 Java MIR .....	78
3.2.4 MIRtoolbox .....	79
3.2.5 MusicMiner .....	81
3.2.6 Tunebot .....	82
4 ANÁLISE COMPARATIVA .....	85
4.1 CRITÉRIOS DE ANÁLISE .....	85
4.2 ANÁLISE .....	87
4.2.1 Soluções Comerciais .....	87
4.2.2 Soluções Acadêmicas .....	91
4.3 DISCUSSÃO DOS RESULTADOS .....	92
4.3.1 Soluções Comerciais .....	93
4.3.2 Soluções Acadêmicas .....	94
5 CONSIDERAÇÕES FINAIS .....	98
REFERÊNCIAS .....	100
APÊNDICE A – Artigo .....	109

## 1 INTRODUÇÃO

Nós ouvimos uma variedade de sons a todo momento e vivemos toda a nossa vida rodeados por eles. Sons de portas abrindo e fechando, dos passos, do ruído dos motores dos automóveis, da chuva e da música. O som não é algo que podemos ver com nossos olhos (MILETTO et al., 2004). Então, o que é som?

O som é a variação da pressão do ar (MULLER, 2007). Sendo assim, a forma de produzir um determinado som depende da maneira como a pressão do ar varia.

Representar o som numericamente é chamado de *digitalização*, ou seja, é representar uma onda sonora (áudio analógico) em código binário (áudio digital). Um processo onde o som é amostrado, quantificado e por fim, codificado, ou seja, transformado em áudio digital (ZUBEN, 2004).

Mas armazenar dados de áudio em formato analógico consome muito espaço (REZENDE et al., 2004), e como solução foi adotado os formatos de compressão de áudio, reduzindo seu tamanho.

Em paralelo, tornou-se possível codificar arquivos de áudio com mais informação do que apenas o nome do arquivo, utilizando os *metadados*. Com os metadados é possível nomear, descrever, catalogar e indicar os direitos de autor de um arquivo de áudio digital. Portanto, um arquivo de áudio digital é composto por metadados e som digitalizado, sendo assim, um dado musical Pacheco e Lopes (2014).

O dado é a representação física de um evento no tempo e espaço que não agrega fundamento, não podendo ser possível entender o que ele representa ou para que ele existe, porém, ao incluir um “significado” no dado e gerar sentido para quem o ouve e ficando claro ou não a que se refere, é gerada a informação (SEMIDÃO, 2013).

Uma informação musical apresenta determinadas especificidades de comportamento na sua produção, objetivação e uso, pois a manifestação da música apresenta-se carregada de características próprias. Portanto, a compreensão completa da música está diretamente ligada com o reconhecimento do contexto histórico e social de sua origem, com a interpretação pessoal e individual do ouvinte, e com os aspectos sonoros que a constituem. Dessa forma, a música tem diferentes significações para cada indivíduo (MICHELS, 1992). Assim, a música é uma expressão humana construída socialmente e objetivada através de sua comunicação oral, registro sonoro ou representação gráfica (BARROS, 2012).

Até o surgimento dos inventos tecnológicos, a música era um meio de comunicação exclusivamente presencial. Com o decorrer do tempo, as técnicas e invenções aplicadas ao

processo de gravação do som foram surgindo e se aperfeiçoando, resultando em aparelhos reprodutores e suportes cada vez mais versáteis e manipuláveis (DAQUINO, 2012). A música se tornou um objeto de consumo universal e extremamente acessível (GOMES, 2015).

Com a Internet, a música ultrapassa os limites físicos da mídia, mergulhando no universo digital. Ela passa a circular livremente pela rede mundial de computadores através do *streaming* (JUNIOR; SEGUNDO, 2008) e a popularização de aplicativos que oferecem mais de 30 milhões de músicas a seus usuários.

Desta forma, a organização da informação, que inclui a sua representação, tem a principal finalidade de possibilitar a recuperação dessa informação, além da sua guarda para a posteridade. A busca por similaridade musical está inserida dentro de um tema de estudos denominado *Music Information Retrieval* (MCLANE, 1996). Sua produção se intensificou com a explosão do interesse em coleções que possuam obras musicais na forma digital, possibilitadas pelo desenvolvimento da compressão de áudio. Os pesquisadores de MIR observam que a motivação maior para essa área de pesquisa é o grande volume de música digital disponível na Internet que, quanto mais cresce, menos possibilita sua recuperação eficiente, visto que estão disponíveis em grande volume, mas sem o tratamento adequado (GOMES, 2015).

Segundo Santini e Souza (2007), a área de MIR conta com profissionais das mais diversas áreas incluídas na questão do tratamento e recuperação da informação musical, mas não apresenta uma ação interdisciplinar, o que prejudica todo o seu processo de comunicação científica, pois não há um periódico ou livro-texto fundador onde pessoas interessadas podem adquirir as bases teóricas e práticas de MIR.

Com exceção de alguns pequenos encontros interdisciplinares, muitos pesquisadores estão apresentando seus resultados para membros das suas próprias disciplinas. A literatura de MIR é difícil de ser localizada, lida e estudada, o que dificulta construir e sustentar uma área de pesquisa respeitável e próspera (SANTINI; SOUZA, 2007).

Diante das dificuldades provenientes da escassa produção científica a respeito do tema, o problema de representação e o processo de construção de sistemas de processamento e recuperação musicais, agrava-se com a necessidade de desenvolvimento de sistemas com estruturas internas, o mais compatível possível com as visões ou desejos dos usuários.

Este trabalho busca reunir informações com o propósito de contribuir com futuros trabalhos, que desejam desenvolver soluções para a busca por similaridade de dados musicais. A intenção é analisar soluções que não necessariamente buscam dados musicais

apenas através do casamento direto de parâmetros de entrada para a busca, como título da música, palavras-chave ou um áudio como parte da música, mas também através do casamento aproximado (ou similar) destes parâmetros.

Portanto, a relevância deste trabalho contribui diretamente, para agregar conhecimento, com o estudo sobre a recuperação da informação de dados musicais, que auxiliarão no desenvolvimento futuro de soluções para busca por similaridade de dados musicais. A pesquisa também tem como objetivo mostrar, de forma clara, o estado da arte sobre a recuperação da informação de dados musicais, identificar formatos de dados musicais e como é feito o armazenamento deles em bancos de dados ou repositórios digitais, apresentar métodos e algoritmos utilizados para busca por similaridade de dados musicais, e, por fim, apresentar e comparar algumas soluções existentes para busca de dados musicais.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo Geral

Este trabalho de conclusão de curso tem como objetivo geral estudar e realizar uma análise comparativa de soluções existentes que não necessariamente buscam dados musicais apenas através do casamento direto de parâmetros de entrada para a busca, mas também através da similaridade destes parâmetros.

### 1.1.2 Objetivos Específicos

Os objetivos específicos necessários para se alcançar o objetivo geral são os seguintes:

- Estudar a fundamentação teórica sobre a recuperação da informação de dados musicais;
- Identificar formatos de dados musicais e como é feito o armazenamento deles em bancos de dados ou outros tipos de repositórios digitais;
- Apresentar métodos e algoritmos utilizados para busca por similaridade de dados musicais;
- Apresentar e comparar algumas soluções existentes para busca de dados musicais.

## 1.2 METODOLOGIA

Esse estudo tem por finalidade realizar um projeto de pesquisa aplicada, uma vez que utilizará conhecimento da pesquisa básica para resolver problemas.

Para um melhor tratamento dos objetivos e melhor apreciação desta pesquisa, observou-se que ela é classificada como pesquisa qualitativa e exploratória. Detectou-se também a necessidade da pesquisa bibliográfica, uma vez que, a pesquisa bibliográfica implica em que os dados e informações necessárias para realização da pesquisa, sejam obtidos a partir de abordagens já trabalhadas por outros autores através de livros, artigos, *surveys*, revistas especializadas, documentos eletrônicos e enciclopédias, entre outras fontes (SILVA; MENEZES, 2005).

Neste trabalho foi feito o uso e coleta de dados de material já publicado, constituído principalmente de teses, dissertações e artigos científicos na busca e alocação de conhecimento sobre recuperação da informação musical.

A primeira etapa consistiu em um levantamento do estado da arte sobre "recuperação da informação musical". A partir desse levantamento, para embasar a pesquisa, foi feito um estudo teórico sobre as tecnologias envolvidas no processo de recuperação da informação musical.

A segunda etapa focou na discriminação resumida de algumas soluções existentes, para busca de dados musicais relacionados ao tema abordado.

A terceira etapa consistiu no detalhamento de critérios e subcritérios a serem utilizados na comparação entre as soluções citadas, que apresentem documentações públicas disponíveis e com informação necessária e suficiente.

Na quarta etapa foi realizado uma análise comparativa entre as soluções escolhidas conforme os critérios e subcritérios estabelecidos na terceira etapa.

Na quinta etapa foi retomado o objetivo proposto, apresentando uma síntese do trabalho, citando os pontos relevantes e os resultados obtidos na análise. Além de apontar ideias para trabalhos futuros encontrados durante o desenvolvimento do trabalho e explanar o aprendizado adquirido.

## 1.3 ESTRUTURA DO TRABALHO

Este trabalho de conclusão de curso foi estruturado baseando-se nos procedimentos para normalização de trabalhos acadêmicos, conforme as regras da Associação Brasileira de Normas Técnicas(ABNT) e documentos disponibilizados pela Biblioteca Universitária<sup>1</sup>

---

<sup>1</sup><http://portal.bu.ufsc.br/normalizacao/>

da Universidade Federal de Santa Catarina(UFSC).

O restante deste trabalho está organizado conforme descrito a seguir. A Seção 2 apresenta o estado da arte sobre a recuperação da informação de dados musicais, a identificação de formatos de dados musicais e como é feito o armazenamento deles em bancos de dados ou repositórios digitais, e a apresentação de alguns métodos e algoritmos utilizados para busca por similaridade de dados musicais.

A Seção 3 apresenta de forma resumida algumas soluções existentes para busca de dados musicais. A Seção 4 apresenta o detalhamento dos critérios de comparação, a análise entre as soluções escolhidas e os resultados obtidos à partir da análise. Por fim, a Seção 5 apresenta a conclusão obtida com o desenvolvimento do trabalho e as ideias encontradas para o desenvolvimento de trabalhos futuros.



## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os conceitos básicos sobre som, quais as representações e estruturas do áudio digital, o que é informação musical e como é feito o seu armazenamento em bancos de dados, por fim, o tema recuperação da informação musical.

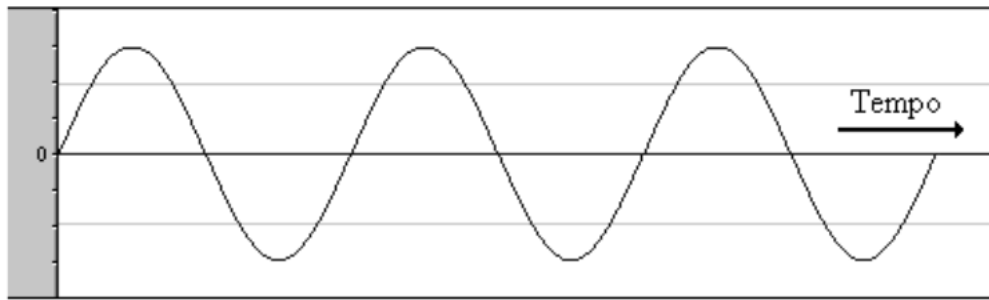
### 2.1 CONCEITOS BÁSICOS SOBRE SOM

Nós ouvimos uma variedade de sons à todo momento e vivemos toda a nossa vida rodeados por eles. Sons de portas abrindo e fechando, dos passos, do ruído dos motores dos automóveis, da chuva e da música. O som não é algo que podemos ver com nossos olhos (MILETTO et al., 2004). Então, o que é o som? Um som é gerado por algum objeto vibratório, em uma repetição periódica de deformações e restaurações (MULLER, 2007). Estas vibrações produzem mudanças de pressão do ar, resultando em regiões locais de ar que são mais densas e outras que são rarefeitas, ocorrendo sucessivamente uma depois da outra e expandindo-se. Estas são chamadas condensações e rarefações. O processo é similar ao que conhecemos quando jogamos uma pedra dentro d'água, a qual produz ondas circulares em sua superfície. Estas ondas de condensações e rarefações são propagadas para dentro do ouvido humano e irão vibrar o tímpano. As vibrações do tímpano são captadas pelas nossas terminações nervosas, de maneira que nós as escutamos como sons. Se os corpos que vibram são diferentes, também será diferente a classe de vibração que produzem. Isto significa que escutamos distintas classes de sons. Outra forma é através de aparelhos chamados *tradutores* (um microfone, por exemplo), que converte as vibrações em corrente elétrica equivalente ao sinal sonoro (ZUBEN, 2004).

Se essa pressão do ar varia de acordo com um padrão repetitivo, dizemos que o som tem uma forma de onda periódica. Se não há um padrão perceptível no som, este é chamado de ruído. Ainda quando as variações na pressão do ar são representadas de forma gráfica, elas podem ser interpretadas como “formas de onda”. Na Figura 1, a representação gráfica de um som mostra as mudanças na pressão do ar conforme a passagem do tempo. Lendo-se o gráfico da esquerda para a direita, quando a linha curva está próxima da parte inferior do gráfico então a pressão do ar é mais baixa, e quando a curva está próxima do topo do gráfico, a pressão do ar aumentou (MILETTO et al., 2004).

Segundo Miletto et al. (2004) e Ferreira (2015), os cinco elementos básicos do som são:

Figura 1 – Representação gráfica, no domínio temporal, de uma forma de onda senoidal


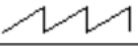



Fonte: (MILETTO et al., 2004)

1. *Altura tonal*: É a “altura” de um som, ou seja, se é alto ou baixo, agudo ou grave. A repetição de uma onda periódica é chamada de ciclo. O número de ciclos dentro do intervalo de um segundo é chamado frequência, medida em hertz (Hz), que é, então, a recíproca do período. Quanto maior o valor em hertz, mais agudo é o som. Dobrando a frequência de um som, este é elevado em uma oitava. Então, é possível dizer que a frequência e a altura tonal estão relacionadas logaritmicamente.
2. *Volume*: A mudança no volume de um som pode ser vista como uma diferença na altura das ondas. A altura de uma onda chama-se “amplitude”. Quanto maior a amplitude, mais forte é o som. Portanto, o volume ou intensidade de um som é determinado pela amplitude.
3. *Timbre*: É o que diferencia dois sons de mesma frequência. De um modo geral, formas de ondas arredondadas produzem um timbre mais suave, enquanto que as formas de ondas ponteadas dão um timbre mais penetrante e estridente.

A Figura 2 mostra três formas de onda básicas, seus timbres característicos e os instrumentos que se assemelham a cada caso.

Figura 2 – Formas de onda simples e seus timbres

Forma de onda	Timbre	Instrumento
<i>Onda senoidal</i> 	<i>suave, doce</i>	<i>flauta, assovio</i>
<i>Onda dente-de-serra</i> 	<i>claro, brilhante</i>	<i>violino, trompete</i>
<i>Onda retangular</i> 	<i>simples, “quente”</i>	<i>clarinete, oboé</i>

Fonte: (MILETTO et al., 2004)

4. *Envolvente da Onda*: É a variação da altura tonal, do volume e do timbre em um transcurso de tempo que vai desde o começo do som até um ponto do tempo onde ele desaparece completamente. Estas mudanças no tempo são o que determina o timbre característico de um objeto, além do seu espectro harmônico.
5. *Duração*: A duração é responsável pelo tempo de emissão de um determinado som. Alguns sons possuem maior ressonância que outros, como um tambor que continua soando por um período de tempo curto, ou os sinos que soam por períodos maiores.

Os sons com vibrações regulares - que são os harmônicos e a fundamental <sup>1</sup> - são considerados sons musicais, enquanto os sons causados por vibrações irregulares - que não são harmônicos - cuja altura tonal não pode, portanto, ser medida, são chamados sons não-musicais. A maioria dos sons usados na música são, por pressuposto, sons musicais.

Á partir desses elementos, segundo Ferreira (2015), temos a composição das propriedades da música, que são:

1. *Harmonia*: Pode ser definida como sendo o agrupamento sonoro. As combinações das notas tocadas simultaneamente de forma coordenada que compõem a harmonia.
2. *Melodia*: É a característica com maior destaque de uma música. Define-se como sendo uma sequência sonora tocada em intervalos irregulares.
3. *Ritmo*: Definido como a sequência de pulsações que se alternam entre sons e silêncios ao longo de um espaço de tempo.

É possível ter uma melhor visualização da relação entre os elementos básicos do som e as propriedades da música na Figura 3.

## 2.2 REPRESENTAÇÃO DO ÁUDIO DIGITAL

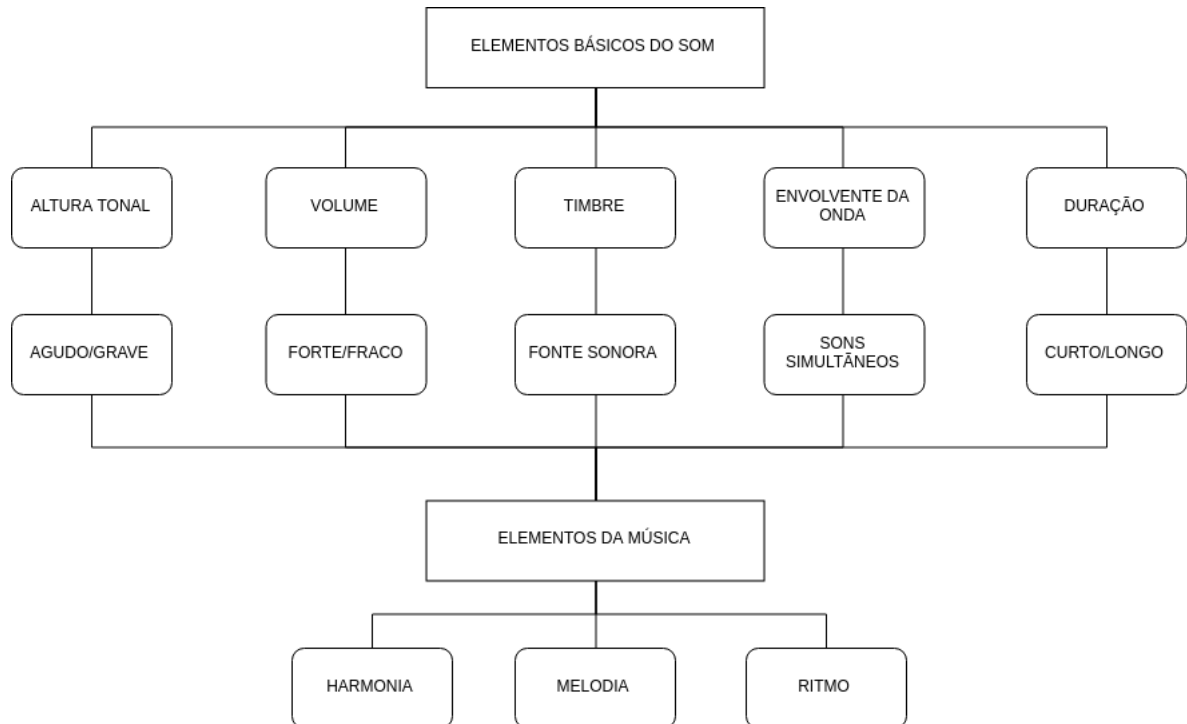
Como comentado anteriormente, o som é a variação da pressão do ar. Sendo assim, a forma de produzir um determinado som depende da maneira como a pressão do ar varia. Para Miletto et al. (2004), o processo de representar numericamente o som é chamado de *digitalização*, ou seja, é representar uma onda sonora (áudio analógico) em código binário (áudio digital).

A digitalização das formas de onda consiste em três etapas realizadas por um circuito chamado *conversor analógico/digital* (A/D) (ZUBEN, 2004): *amostragem*, *quantização* e

---

<sup>1</sup>Por definição, o primeiro harmônico corresponde à frequência fundamental, o segundo harmônico ao primeiro harmônico e assim por diante (MULLER, 2007).

Figura 3 – Relação entre elementos do som e da música



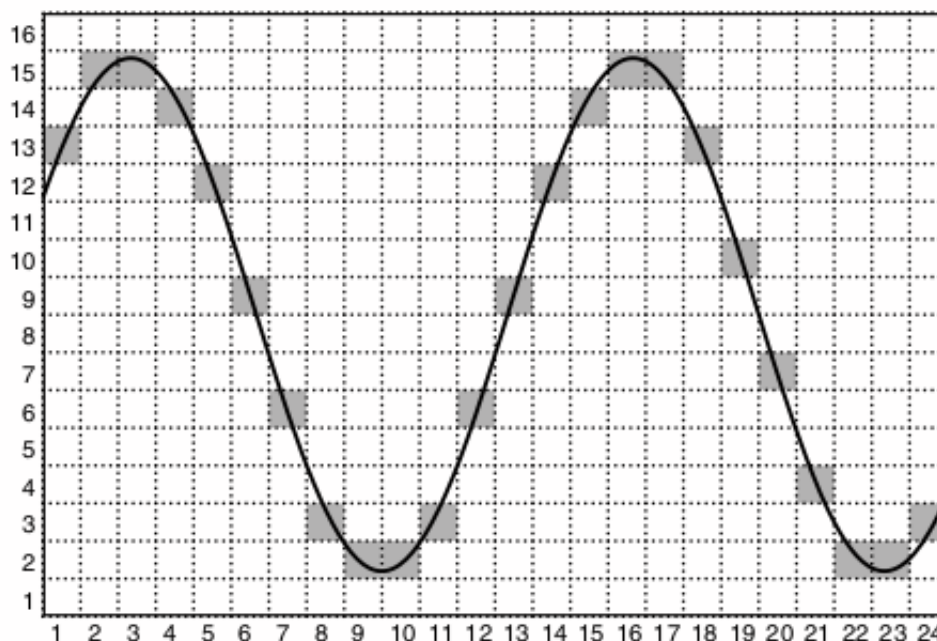
Fonte: (FERREIRA, 2015), adaptado pela autora

*codificação*. Na primeira etapa, a forma de onda é lida ou amostrada em curtos intervalos de tempo uniformes (*samples*). Na segunda etapa, o valor da forma de onda em cada ponto amostrado é restrito ou quantificado para um conjunto discreto de valores. Por fim, na terceira etapa é feita a codificação, traduzindo os valores para código binário. É uma transformação com perdas, ou seja, se perde informação neste processo (ver Figura 4). Para podermos ouvir as informações transformadas em linguagem digital pelo conversor A/D é preciso realizar o processo inverso utilizando um *conversor digital/analógico* (D/A) (ZUBEN, 2004).

Inicialmente, armazenar dados de áudio em formato analógico discretizado consome muito espaço (REZENDE et al., 2004) e, como solução, foi adotada a codificação MIDI (*Musical Instrument Digital Interface*). Essa codificação é uma representação numérica do som, sendo usada atualmente como o formato de intercâmbio de música digital simbólico mais comum para possibilitar a transferência de informações entre instrumentos musicais e computadores (MULLER, 2007). O tamanho desses arquivos tende a ser bem reduzido, já que a codificação de eventos musicais pode ser bem compacta.

Embora isso seja satisfatório em algumas situações, arquivos MIDI (extensão .mid ou .smf) basicamente codificam apenas instruções referentes a notas e durações e não a informação sonora propriamente dita (IAZZETTA; KON, 1998). Ou seja, seu código é

Figura 4 – Forma de onda analógica (curva preta) e representação digitalizada (as amostras digitalizadas são indicados pelas caixas cinzentas). Neste exemplo, a forma de onda é amostrada em 24 pontos e a quantização dos valores emprega um esquema de codificação de 4 bits (16 valores possíveis)



Fonte: (MULLER, 2007)

composto por instruções para um sintetizador correspondente a eventos musicais como soar uma nota, silenciar uma nota, mudar o andamento ou o tom da música, etc. O sintetizador, por sua vez, interpreta este código e o executa, podendo usar vários timbres de instrumentos simultaneamente (MILETTO et al., 2004). Por isso, o resultado sonoro é totalmente dependente da qualidade e das possibilidades oferecidas por esse aparelho (IAZZETTA; KON, 1998).

Com o mesmo objetivo, os formatos *Wave*, da Microsoft (extensão .wav), e *AIFF*, da Apple (extensão .aiff ou .aif) são do tipo áudio digitalizado sem compressão. Baseiam-se na codificação PCM (*Pulse Code Modulation*), onde a própria onda sonora é representada como uma sucessão de números correspondentes às amplitudes do sinal medidas a uma frequência constante. Ou seja, armazenam dados de um processo de digitalização simples. A diferença para o formato MIDI é que esse tipo de codificação origina um grande volume de dados e exige muito espaço para armazenamento. Sua única vantagem é a reprodução fiel do áudio se o arquivo for gravado com qualidade de CD (MILETTO et al., 2004).

Outro recurso é a compressão de dados, que é feita através de programas ou *hardware* específicos que compactam os arquivos de áudio, reduzindo o seu tamanho antes de serem enviados. Ao chegar ao seu destino, esses arquivos são descompactados e, em

seguida, tocados. Entretanto, compressão, nesse contexto, é sinônimo de perda de qualidade: quanto maior a compressão, maior também a quantidade de informação que se perde (IAZZETTA; KON, 1998). Apesar disso, certos padrões de compressão já permitem a transmissão de arquivos de áudio de boa qualidade em tempo real, o que resulta na existência de uma quantidade equivalente de formatos de arquivo para a compressão do som digitalizado (MILETTO et al., 2004).

Segundo Miletto et al. (2004), os formatos de arquivo mais utilizados para a compressão de som são:

- MPEG Layer 3 (extensão .mp3);
- Advanced Audio Coding (extensão .aac);
- Windows Media Audio, da Microsoft (extensão .wma);
- Real Audio, da RealNetworks (extensão .ra);
- Sun Audio, da Sun (extensão .au).

Em paralelo, quando o áudio foi do analógico para o digital, tornou-se possível codificar arquivos de áudio com mais informação do que apenas o nome do arquivo, utilizando-se dos chamados *metadados*.

Metadados podem ser usados para nomear, descrever, catalogar e indicar os direitos de autor de um arquivo de áudio digital. Como diferentes formatos de áudio digitais foram desenvolvidos, foi acordado que um local padronizado e específico seria reservado dentro dos arquivos digitais, onde estas informações pudessem ser armazenadas (PACHECO; LOPES, 2014). Existem padrões diferentes de metadados para finalidades distintas de informações. Para o áudio digital, o padrão mais utilizado é o de *tags* ID3<sup>2</sup> (abreviação de *IDentify a MP3*).

O ID3 começou a ser idealizado a partir de 1996 por Eric Kemp<sup>3</sup>. Em sua primeira versão, o ID3 limitava-se apenas a 128 KB de informação, e essas informações também eram limitadas quanto ao número de caracteres. Já na segunda versão, idealizada a partir de 1998, o padrão tornou-se mais completo, permitindo que mais informações pudessem ser adicionadas, já que o limite passou a ser de 256 MB (FERREIRA, 2015).

Na Tabela 1 é possível verificar as diferenças com relação aos metadados considerados pelas versões do ID3.

---

<sup>2</sup>Embora referido como um padrão, o termo se aplica apenas no sentido de palavra, já que nenhum organismo de normalização foi envolvido na sua criação, e nenhuma dessas organizações lhe deu esse estatuto de aprovação formal (PACHECO; LOPES, 2014).

<sup>3</sup><http://id3.org/>

Tabela 1 – Diferenças entre as versões ID3

ID3v1	ID3v2
128 B	256 MB
Título	Título
Artista	Artista
Álbum	Álbum
Ano	Ano
Comentário	Comentário
Gênero	Gênero
	Compositores
	Letra da música
	Capa do álbum

Fonte: (FERREIRA, 2015), adaptado pela autora

Segundo Pacheco e Lopes (2014), metadados podem ser armazenados de duas formas:

- Interno (ou incorporado): os metadados são transportados como parte do arquivo de áudio digital;
- Externo: os metadados podem ser unidos com o conteúdo quando a informação é transferida ou referenciada.

Um Banco de Dados(BD) normalmente armazena os metadados externamente, mas pode ser projetado para suportar abordagens para o armazenamento interno de metadados. Cada qual apresenta vantagens e desvantagens, como pode ser visto na Tabela 2.

Existem diversos *softwares* que editam as *tags* ID3, porém, na maioria das vezes esses *softwares* disponíveis não permitem a edição completa dos dados de descrição do áudio.

Desta forma, podemos afirmar que um arquivo de áudio digital é composto por metadados e som digitalizado, sendo assim, um dado musical.

## 2.3 INFORMAÇÃO MUSICAL

Para Setzer (2001), dado é uma sequencia de números, portanto, um texto, fotos, figuras, sons gravados e animação são dados, pois todos podem ser amostrados, quantifi-

Tabela 2 – Diferenças entre as formas de armazenamento de metadados

<b>Armazenamento Interno</b>	<b>Armazenamento Externo</b>
Não permite a gestão de todos os metadados em um só lugar	Procura e gestão mais eficiente dos metadados
Cria redundância impedindo a normalização	Evita redundância através da normalização
Os metadados estão sempre disponíveis e podem ser manipulados no local	Os arquivos sozinhos não transportam os metadados

Fonte: Elaborado pela autora

cados e codificados. Um dado é necessariamente uma entidade matemática e, desta forma, é puramente sintático. Isto significa que os dados podem ser totalmente descritos através de representações formais, estruturais e obviamente ser armazenados em um computador e processados por ele.

O dado é a representação física de um evento no tempo e espaço que não agrega fundamento, não sendo possível entender o que ele representa ou para que ele existe, se somente for disponibilizado para alguém ou para o tempo e espaço, por alguém ou por um evento. Entretanto, ao incluir um “significado” no dado e gerar sentido para quem o ouve e ficando claro ou não a que se refere, é gerada a informação (SEMIDÃO, 2013).

O autor Setzer (2001) expõe que a informação é uma abstração informal (isto é, não pode ser formalizada através de uma teoria lógica ou matemática) que está na mente de alguém, representando algo significativo para essa pessoa. A frase “Paris é uma cidade fascinante” é um exemplo de informação desde que seja lida ou ouvida por alguém, desde que "Paris" signifique para essa pessoa a capital da França (supondo-se que o autor da frase queria se referir a essa cidade) e "fascinante" tenha a qualidade usual e intuitiva associada com essa palavra.

A informação pode ser armazenada em um computador, ou melhor, a sua representação em forma de dados e não a informação propriamente dita. Essa representação pode ser transformada pela máquina, como na formatação de um texto, mas não pode mudar o significado, já que ela depende de uma pessoa que possui a informação. Os dados são sempre incorporados por alguém como informação, porque os seres humanos buscam constantemente por significação e entendimento.

A partir daí, a autora Barros (2012) expõe que um dado musical é resultado de um processo de significação social. Assim, a música é uma expressão humana construída socialmente e objetivada por meio de sua comunicação oral, registro sonoro ou representação



gráfica.

Para Almeida (2007), a obra musical é efêmera e abstrata, pois só se concretiza no momento de cada interpretação, na execução da música.

Berger e Luckmann (2014) afirmam que:

A expressividade humana é capaz de objetivações, isto é, manifesta-se em produtos da atividade humana que estão ao dispor, tanto dos produtores quanto dos outros homens, como elementos que são de um mundo comum (BERGER; LUCKMANN, 2014).

Para Angeles, McKay e Fujinaga (2010), Pinheiro e Loureiro (1995), a informação é o que se acrescenta a uma representação. Segundo os autores, recebemos informação se o que conhecemos é alterado. Informação é o que logicamente justifica alteração ou reforço de uma representação, ou de um estado de coisas. As representações podem ser explícitas (como em um mapa ou em uma proposição), ou podem estar implícitas no estado de atividade dirigida do receptor.

A informação musical apresenta determinadas especificidades de comportamento na sua produção, objetivação e uso, pois a manifestação da música apresenta-se carregada de características próprias. Michels (1992) explica que a música contém dois elementos: o *material acústico* e a *ideia intelectual*, sendo que tais elementos não se encontram justapostos, mas sim se combinam para formar uma imagem unitária. Portanto, a compreensão completa da música está diretamente ligada com o reconhecimento do contexto histórico e social de sua origem, com a interpretação pessoal e individual do ouvinte, e com os aspectos sonoros que a constituem. Dessa forma, a música tem diferentes significações para cada indivíduo.

Lima e Santini (2006) afirmam que:

A música é um produto social e simbólico de grande importância nas diferentes formações culturais, principalmente se considerarmos a sua capacidade de criar vínculos afetivos e cognitivos entre as pessoas (LIMA; SANTINI, 2006).

A compreensão da música como informação é ainda bastante recente. O estudo mais significativo e considerado, dentro da literatura especializada, como pioneiro na conceituação e estudo da música como fonte de informação é o de Alexander McLane. Em 1996, o autor publicou em um capítulo do *ARIST (Annual Review of Information Science and Technology)* o artigo intitulado *Music as Information*, onde formaliza a música como informação.

## 2.4 ARMAZENAMENTO DA INFORMAÇÃO MUSICAL

Até o surgimento dos inventos tecnológicos, a música era um meio de comunicação exclusivamente presencial. Apesar das formas de registros, a exemplo das partituras, possibilitarem a execução de uma obra em diferentes momentos e lugares, a reprodução do que ali estava representando nunca seria a mesma. Com o decorrer do tempo, as técnicas e invenções aplicadas ao processo de gravação do som foram surgindo e se aperfeiçoando, resultando em aparelhos reprodutores e suportes cada vez mais versáteis e manipuláveis (DAQUINO, 2012). Especialmente após o desenvolvimento das coleções em rede na *Web*, com formatos de arquivos compactados e custos decrescentes de armazenamento de arquivos na forma digital, a música se tornou um objeto de consumo universal e extremamente acessível (GOMES, 2015).

Os dispositivos de armazenamento musical são divididos em dois grandes grupos: *analógicos* e *digitais* (ANDRADE; CRISPIM; OGIBOWSKI, 2008). Os analógicos são antecessores dos digitais e foram o meio tecnológico dominante em boa parte do século XX (ZUBEN, 2004). O primeiro invento significativo foi o fonógrafo, patenteado por Thomas Edison em 1877. Dez anos mais tarde, em 1887, surgiu o gramofone, tendo uma capacidade maior de armazenamento e reprodução das músicas (MARCHI, 2005).

Dentre as invenções mais importantes para o armazenamento e a reprodução sonora analógica está o disco de vinil, lançado em 1948, comumente conhecido como LP. Era um disco com rotação por minuto mais demorada, o que permitia aumentar a capacidade de armazenamento da informação na superfície do vinil. Em seguida, entre as décadas de 60 e 70, com a evolução dos cartuchos 8-track (pioneiros em armazenar dados musicais em fitas magnéticas), o lançamento da fita cassete ou *compact cassette* (MARCHI, 2005) era basicamente, dois carretéis, a fita magnética e todo o mecanismo de movimento da fita, alojados em uma caixa plástica (ANDRADE; CRISPIM; OGIBOWSKI, 2008).

No armazenamento analógico, as formas de onda dos sinais elétricos emitidos do aparelho eram registradas similarmente, isto é, de maneira análoga, pelas partículas magnéticas encontradas na fita. No momento da reprodução, os sinais magnéticos impressos na fita são interpretados analogamente como diferenças de voltagem, isto é, sinais elétricos. Como o nível do sinal elétrico era muito baixo, utilizava-se um amplificador para que a variação de voltagem seja suficiente para mover os cones dos alto-falantes. Dizemos que esse processo é uma gravação analógica, pois a forma de onda do sinal gravado é análoga à forma de onda do sinal original captado (ZUBEN, 2004).

Porém, a partir da década de 80, com a intensificação do uso de *hardwares* e *softwa-*

res, surge um dos primeiros meios de armazenamento digital: o CD (*Compact-Disc*). Ele acabou por se tornar um dos meios de armazenamento de dados musicais mais populares das décadas seguintes. Além de quebrar paradigmas na época, o CD foi inspiração para o desenvolvimento de outros meios de armazenamento como os DVDs e os discos de Blu-ray (MARCHI, 2005).

Entretanto, surgiu a necessidade de disponibilizar informações nos dispositivos de armazenamento. Na era analógica, esses recursos estavam associados à capa e/ou contracapa. Todavia, com a mudança do paradigma para digital, necessitou-se que essas informações pudessem ser disponibilizadas digitalmente, porém, o CD de áudio não inclui em sua estrutura, por exemplo, o nome do disco ou o nome de suas faixas. Sendo assim, surgiram as bases de dados de CDs, que visam prover informações dos mesmos quando esses são utilizados por sistemas de mídia modernos (ANDRADE; CRISPIM; OGIBOWSKI, 2008).

Com o surgimento dos arquivos digitais de áudio, as músicas se desvincularam do suporte físico (CD) e passaram a ser vistas isoladamente. Esses tipos de arquivos requerem um espaço considerável para seu armazenamento (chegando a ocupar dezenas de megabytes em disco), o que propiciou o surgimento de um formato mais compacto: O MP3 (*Moving Picture Experts Group-1 Layer 3*).

O grande desenvolvimento tecnológico das redes de compartilhamento de arquivos contribuíram para uma maior aceitação de formato de áudio MP3 (ANDRADE; CRISPIM; OGIBOWSKI, 2008), por ser facilmente transportado em qualquer bolso ou mochila e pela sua longa vida útil, além do aumento gradativo de armazenamento com o passar dos anos (MARCHI, 2005). Sua invenção, propiciou a popularização de eletrônicos com portas USB e o surgimento de cartões de memória *microSD*, prometendo maior capacidade, durabilidade e clareza sonora (MARCHI, 2005).

Com a Internet, a música ultrapassa os limites físicos da mídia - mergulhando no universo digital - e passa a circular livremente pela rede mundial de computadores através do *streaming*, que tomou forma no final da década de 80 e começou a se desenvolver na década de 90, com a evolução dos SGBDs (*Sistemas de Gerenciamento de Banco de Dados*) e o surgimento dos BDOOs (*Bancos de Dados Orientado a Objetos*) (JUNIOR; SEGUNDO, 2008), que possibilitaram o armazenamento de multimídias, assim como a popularização de aplicativos móveis, conhecidos normalmente por *apps*, que oferecem mais de 30 milhões de músicas a seus usuários, como por exemplo, "Spotify"<sup>4</sup> e "Deezer"<sup>5</sup>.

Ainda, para uma eficiente recuperação de informação é preciso que o conjunto dos

---

<sup>4</sup><https://www.spotify.com/br/>

<sup>5</sup><https://www.deezer.com/br/>

arquivos musicais estejam representados de forma adequada, contemplando as possíveis informações que o usuário busca (FERREIRA, 2015).

## 2.5 RECUPERAÇÃO DA INFORMAÇÃO MUSICAL

A área de pesquisa denominada *Music Information Retrieval* (MIR - *Music Information Retrieval*) ou Recuperação da Informação Musical (RIM), tradução literal incorporada pela corrente da área no Brasil, é definida, de acordo com Futrelle e Downie (2002), como:

[...] uma agenda de pesquisa que, de forma geral, pretende desenvolver formas de gestão de coleções de obras musicais para preservação, busca, acesso e outros usos (FUTRELLE; DOWNIE, 2002).

A agenda de pesquisas sobre a MIR intensificou sua produção recentemente com a explosão do interesse em coleções em rede que contenham obras musicais na forma digital, possibilitadas pelo desenvolvimento das citadas técnicas de compressão de áudio. Os pesquisadores de MIR observam que a motivação maior para essa área de pesquisa é o grande volume de música digital disponível na Internet que, quanto mais cresce, menos possibilita sua recuperação eficiente, visto que estão apenas disponíveis em grande quantidade, mas sem o tratamento adequado (GOMES, 2015).

A área de MIR conta com profissionais das mais diversas áreas inclusas na questão do tratamento e recuperação da informação musical como apresentado na Tabela 3, traduzida por Santini e Souza (2007).

A origem de MIR não apresenta uma ação interdisciplinar, o que prejudica todo o seu processo de comunicação científica. Como pontua Santini e Souza (2007):

(...) não há uma sociedade (inter)disciplinar de MIR; um periódico ou livro-texto fundador onde pessoas interessadas podem adquirir as bases teóricas e práticas de MIR. Com exceção de alguns pequenos encontros interdisciplinares, muitos pesquisadores estão apresentando seus resultados para membros das suas próprias disciplinas. A literatura de MIR é difícil de ser localizada, lida e estudada, o que dificulta construir e sustentar uma área de pesquisa respeitável e próspera (SANTINI; SOUZA, 2007).

A escassa produção científica a respeito do tema e as características impostas pelas músicas acarretam certas dificuldades para sua representação.

Como elucidado no final da subseção 2.1 desta seção, o ponto de partida para os estudos sobre a música como fonte de informação e tratamento, representação e recuperação foi dado em 1996 por Alexander McLane. Neste mesmo período já era possível identificar o desenvolvimento de tecnologias de compressão de arquivos digitais de música

Tabela 3 – Comunidades de MIR

<b>Comunidade</b>	<b>Tipo(s) de instituição</b>	<b>Área de pesquisa</b>
Ciência da Computação, Recuperação da Informação	Acadêmica, Comercial	Representação, Indexação, Recuperação, Aprendizado de máquina, Design de interface de uso
Engenharia de áudio, Processamento de sinais digitais	Acadêmica, Comercial	Compressão, detecção de critério, Localização de tom, Aprendizado de máquina, Classificação, Análise musical
Musicologia, Teoria Musical	Acadêmica	Representação, Análise musical
Ciência da Informação, Biblioteconomia	Bibliotecas, Acadêmica	Representação, Metadados, Estudos de usuário, Classificação, Direitos de propriedade intelectual, Design de interface de uso
Ciência Cognitiva, Psicologia, Filosofia	Acadêmica	Representação, Percepção, Estudos de usuário, Ontologia
Direito	Governamental, Profissionais da lei, Acadêmica	Direitos de propriedade intelectual

Fonte: (SANTINI; SOUZA, 2007 apud FUTRELLE; DOWNIE, 2002)

para transmissão na Internet e a popularização da Internet no mundo (SANTINI; SOUZA, 2007).

Em seu estudo, McLane direciona sua discussão para os grandes problemas relacionados à representação de documentos de música e à recuperação destes documentos. Ele analisa aspectos significantes da música – sua notação e seu som –, e propõe algumas ideias para sistemas de recuperação de música e formaliza a música como informação segundo três visões: *visão subjetiva*, *visão objetiva* e *visão interpretativa*. Segundo o autor, as necessidades dos vários tipos de análises musicais são tão diversas que é preferível considerar três “visões” sobre a representação da obra musical.

Em resumo adaptado, Santini e Souza (2007) apresentam as principais característi-

cas das visões estabelecidas por McLane (1996).

- A visão *subjetiva* da informação musical se faz por meio do uso do esquema de notação para representação da informação musical. A subjetividade se dá porque a escolha de elementos de notação geralmente representa uma obra em “contexto-dependente”. Sendo assim, a decisão da notação pode incluir ou excluir aspectos particulares da obra.
- A visão *objetiva* está vinculada a audição e ao momento da execução musical. Um som gravado pode ser identificado como visão objetiva da obra musical. A sonoridade se caracteriza como objetiva por não se configurar como uma representação, mas como a obra em sua essência. O som musical uma vez gravado torna-se fixo e não está mais sujeito a variações editoriais e de performance. Segundo McLane, esta visão pode ser considerada a mais completa representação da música, ao passo em que inclui as facetas tom, tempo, harmonia, editorial e timbre.
- A visão *interpretativa* é realizada através da análise de alguns aspectos da obra, englobando informações que não são diretamente dependentes do documento. Entram nessa categoria classificações e esquemas analíticos que elucidam características como o gênero musical e avaliações críticas.

De acordo com Cruz (2014):

Dentre as visões propostas por McLane (1996), a interpretativa possui uma característica interessante porque permite a independência formal do documento musical em relação ao suporte que o contém, assim como foi possível na informação textual (CRUZ, 2014).

Segundo McLane (1996), Santini e Souza (2007), a representação da informação musical pode abranger as três visões apresentadas dependendo das necessidades de informação da comunidade usuária. De acordo com tradução das autoras, a conclusão de McLane seria a de que:

Ambas as escolhas sobre a visão da representação da música e o grau de complementação da representação de uma obra depende da necessidade de informação do usuário. A recuperação de informação é um processo interativo que depende do conhecimento do usuário e do nível de complexidade da informação desejada. No caso da necessidade da simples identificação de uma obra musical, onde a informação bibliográfica não é unicamente suficiente, pode-se limitar a uma visão subjetiva envolvendo um subconjunto relativamente pequeno de elementos notados de uma obra, frequentemente o tom inicial de uma frase melódica. A representação tonal pode ser de forma tal que provavelmente o usuário espera e está apto para formular a indagação usando a mesma terminologia, ou

pelo menos uma que é traduzível na forma de representação (SANTINI; SOUZA, 2007 apud MCLANE, 1996).

Sendo assim, percebe-se que a recuperação da informação da música depende tanto da complexidade e da forma como a informação é representada, como do nível de conhecimento prévio do usuário. Para Santini e Souza (2007): “Quanto menor o conhecimento do usuário, maior a necessidade de diferentes formas de representação. Cada visão da representação da música, demonstrada por McLane, não é suficientemente isolada para identificar uma obra.”.

Outro autor presente nos estudos relacionados à representação e recuperação de informação musical e um dos representantes da área de MIR é o professor J. Stephen Downie da Universidade de Illinois, nos Estados Unidos. Downie escreveu, em 2003, outro artigo tido como marco no estudo da informação musical, intitulado *Music Information Retrieval* também em um capítulo do ARIST. No trabalho em questão, Downie (2003) examina a multidisciplinaridade da área de MIR, identifica e explica alguns problemas relacionados à questão da representação e recuperação da informação musical. Para isso, Downie (2003) resume a questão em quatro grandes desafios a serem enfrentados pelos pesquisadores da área:

1. Considerar permanentemente as diferentes formas de representação da música, o que caracteriza o *desafio multirepresentacional*. O copyright faz parte deste desafio.
2. Cada época histórica e cada formação cultural criam modos próprios e singulares de se expressar através da música. A música transcende as fronteiras culturais e temporais. A ampla variedade de expressões musicais coloca em evidência o *desafio multicultural*.
3. Compreender e responder às diferentes formas de interação individual com a música e com os sistemas de MIR constitui o *desafio multierperimental*.
4. Maximizar os benefícios de ter uma comunidade multidisciplinar de pesquisadores, enquanto minimiza a desvantagem inerente, representa o *desafio multidisciplinar*.

O desafio multirepresentacional é dividido em sete facetas a serem consideradas na descrição da música e que representam a estrutura musical (DOWNIE, 2003): tonal, temporal, harmônica, de timbre, editorial, textual e faceta bibliográfica, sendo as quatro primeiras relativas a aspectos sonoros da música com formas gráficas de representação em figuras rítmicas ou notações musicais, enquanto que as três últimas são representadas na forma gráfica e dizem respeito às informações de produção, intérprete, compositor, copyright, data de produção e outras (BARROS, 2012).

Apesar de completas, a interação dessas facetas resulta em um complexo tratamento da informação musical, visto que cada faceta citada possui, por si só, uma complexidade

inerente e sofre um tipo de representação enquanto produto. As autoras Santini e Souza (2007) resumem a problemática multirepresentacional da seguinte forma:

A complexa interação entre as facetas da música - tonal, temporal, harmônica, de timbre, editorial, textual e bibliográfica – evidencia um dos principais problemas de MIR: o desafio multirepresentacional. A escolha da representação da música – se baseada em símbolos, áudio ou ambos – adiciona-se a diversas questões: como, por exemplo, cada escolha determina a tecnologia, a organização, a recuperação e a interface entre requisitos e capacidades dos sistemas (SANTINI; SOUZA, 2007).

Sendo assim, apesar de possuir as facetas estabelecidas por (DOWNIE, 2003), a estrutura da música incorpora elementos extras que nos permitem defini-la como um objeto informacional mais complexo. Para Cruz (2014) a estrutura musical:

[...] incorpora elementos adicionais que permitem defini-la como um objeto informacional musical mais amplo, dotado de conteúdo – atributos internos e metadados descritivos – e, de contexto – associações com outros objetos musicais e não musicais, e com situações ou eventos em que este objeto musical está inserido (CRUZ, 2014).

O segundo desafio (multicultural) nasce da condição inerente à música de ser uma objetivação de algo extremamente subjetivo: a *expressão humana*. Sendo assim, sofre a interferência de uma grande variedade de fatores, da cultura vigente no momento da produção musical e da localização geográfica desta produção.

O desafio multiexperimental diz respeito à percepção da música como experiência individual ou coletiva capaz de causar diferentes reações em diferentes momentos e situações, de cada mente e humor individual. Neste caso, ouvir uma música gravada funciona como “ajudar a memória” que traz a tona experiências prazerosas ou dolorosas relacionadas a uma música em especial (DOWNIE, 2003; SANTINI; SOUZA, 2007). As variações de pessoa para pessoa na forma de apropriação, apreciação e nos tipos de experiências emocionais que a música evoca demonstram de maneira pragmática o desafio multiexperimental.

O quarto e último desafio estabelecido por Downie (2003) é o desafio multidisciplinar. Como citado anteriormente, a diversidade intelectual da comunidade de pesquisadores de MIR é, ao mesmo tempo, uma vantagem e uma adversidade. A heterogeneidade das visões de mundo das disciplinas apresenta um problema particular. Cada disciplina traz suas crenças, práticas, questões de pesquisa e paradigmas de avaliação (DOWNIE, 2003). De acordo com Futrelle e Downie (2002), não há uma aceitação comum dos objetivos, técnicas e resultados obtidos nas pesquisas referentes à informação musical.

Percebe-se, portanto, que, se por um lado, a ação multidisciplinar dos pesquisadores envolvidos com o tema possibilita o surgimento de diversos avanços tecnológicos e que a cada dia são divulgadas novas soluções para o tratamento de conteúdos musicais, com



algoritmos mais sofisticados, novas formas de indexação de músicas, novos tipos de interfaces de áudio e novas formas de representação musical, em contrapartida, é notável a dificuldade de comunicação entre esses resultados. Nota-se ainda a dificuldade de identificação desses conteúdos musicais, porque a música é complexa e possui um leque de propriedades que possibilitam abordagens, às vezes, contraditórias (CRUZ, 2014).

A análise da informação musical para sua representação apresenta complexidades, pois exige diferentes técnicas de extração de informações para distintas formas de apresentação (DOWNIE, 2003).

### 2.5.1 Métodos e Algoritmos para Recuperação da Informação Musical

Um dado musical geralmente é representado por um conjunto de características extraídas do conteúdo da música. Isto porque dados musicais são basicamente arranjos multidimensionais de valores derivados de vários sensores, que é uma representação limitada para definir a semântica do dado. Neste sentido, dados complexos raramente são comparados diretamente. Em vez disso, o conteúdo do dado (ou uma faceta do conteúdo) é analisado por meio de algoritmos especializados de análise, extraíndo um conjunto de características que descrevem numericamente o dado (KASTER, 2012). Desta forma, as aplicações que lidam com dados complexos, como dados musicais, requerem a realização de consultas por similaridade, ou seja, consultas que realizam busca por objetos da base que sejam similares a um objeto de consulta, de acordo com uma certa medida de similaridade (BARIONI, 2006).

A Sociedade Internacional de Recuperação da Informação Musical (ISMIR<sup>6</sup>, na sigla em inglês) abriga, durante sua conferência anual, o MIREX<sup>7</sup>, com o objetivo de estabelecer métodos para a avaliação e comparação das aplicações atuais de recuperação musical. Nesta espécie de competição, pesquisadores inscrevem algoritmos que realizam diferentes tarefas da área de recuperação da informação musical, como classificação automática de gênero musical, identificação automática da pulsação, extração de melodia a partir de arquivos áudio, entre outros.

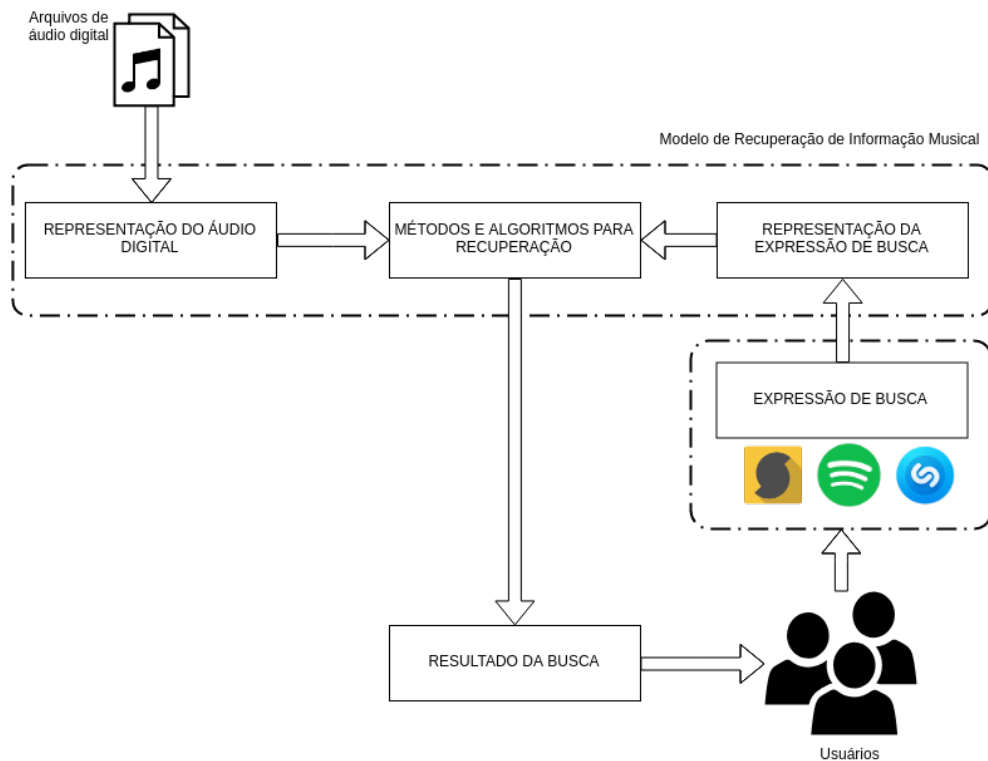
Um sistema de recuperação de informação musical atua, portanto, como um ambiente mediador da comunicação entre as necessidades informacionais dos usuários e o conjunto de arquivos de áudio. Na Figura 5 é possível compreender melhor o processo de recuperação de informação. Dadas as características de um sinal de áudio, pode-se definir métricas de similaridade para comparar músicas. Embora exista uma grande variedade de

---

<sup>6</sup><https://transactions.ismir.net/>

<sup>7</sup>[http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME)

Figura 5 – Representação do modelo de recuperação de informação musical



Fonte: (FERREIRA, 2015), adaptado pela autora

funções de distância disponível na literatura, não existe um método que determine, de um modo geral, qual deve ser a melhor função de distância a ser utilizada em cada caso. A escolha ou definição de uma função de distância é uma tarefa que depende muito da análise das características específicas do domínio dos dados a serem manipulados (BARIONI, 2006).

O foco deste trabalho não é aprofundar o conhecimento nos métodos e algoritmos utilizados para recuperação da informação musical. Assim sendo, os principais métodos e algoritmos disponíveis na literatura são descritos aqui de forma breve.

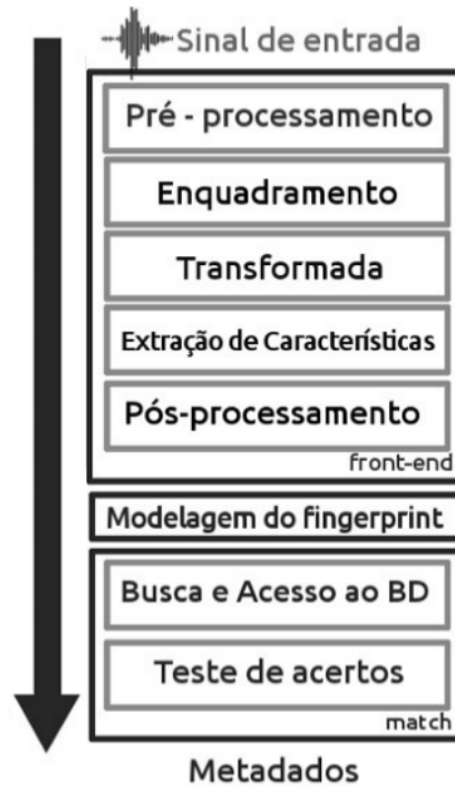
#### 2.5.1.1 Audio Fingerprint

*Audio Fingerprint* é como uma assinatura única de uma música, contendo um sumário de suas características que resume uma gravação de áudio. Cano et al. (2005) que definem que uma *fingerprint* de áudio é um resumo digital, que pode ser utilizado para identificar uma amostra ou localizar rapidamente itens semelhantes em uma base de dados de áudio, independente do nível de compressão, distorção ou interferência no canal de transmissão.

Esse sistema pode ser separado em dois processos fundamentais: extração do *finger-*

*print* (*frontend*) e o algoritmo de comparação (*match*). Estes, por sua vez, são divididos em até oito etapas, conforme mostrado na Figura 6.

Figura 6 – Diagrama de identificação de áudio



Fonte: (CARREIRA, 2015)

No processo de *frontend* há uma sequência de métodos relevantes até a criação de fato do *fingerprint*. Segundo Cano et al. (2005), o primeiro passo é o pré-processamento, onde o som é digitalizado (se necessário) e convertido em um formato genérico, por exemplo, para o formato de 16 bits PCM<sup>8</sup>.

O enquadramento adapta a amostragem convertida para o que será utilizado e Cano et al. (2005) definem que:

[...]o sinal é dividido em quadros de um tamanho comparável à velocidade de variação dos eventos acústicos subjacentes. O número de quadros calculados por segundo é chamado de taxa de quadros. Uma função de janela cônica é aplicada a cada bloco para minimizar as descontinuidades no início e no final. A sobreposição deve ser aplicada para garantir robustez ao deslocamento (ou seja, quando os dados de entrada não estão perfeitamente alinhados com a gravação usada para gerar a *fingerprint*) (CANO et al., 2005).

Em seguida, precisa-se ter o áudio no domínio da frequência (BUNNELL, 1996a). Assim, transformações adicionais são aplicadas que convertem esses sinais em função da

<sup>8</sup>*Pulse Code Modulation*: método usado para representação digital de sinais analógicos.

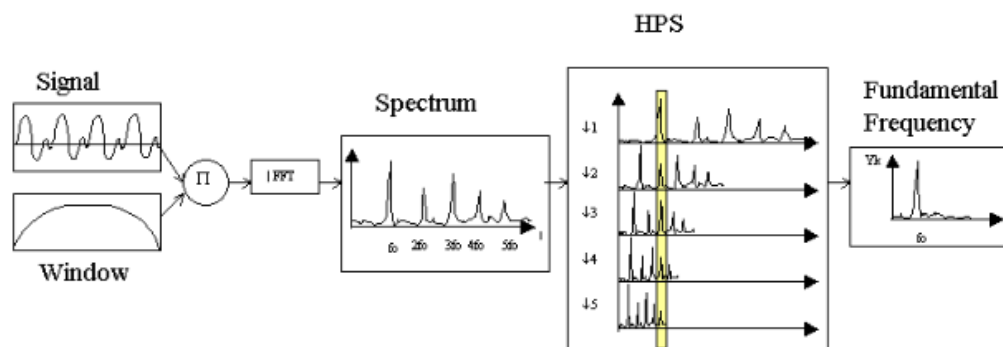
frequência, para que o algoritmo de reconhecimento possa realizar uma melhor classificação dentro das características desejadas (SANTOS, 2011).

Cano et al. (2005) citam algoritmos de Transformadas para facilitar a compressão eficiente, a remoção de ruídos e o processamento subsequente:

- Transformada de Karhunen-Loève;
- Transformada Rápida de Fourier;
- Transformada de Walsh-Hadamard;
- *Modulated Complex Lapped Transform (MCLT)*;
- Transformada Wavelet.

Santos (2011) cita um algoritmo prático conhecido como o *espectro do produto harmônico*, ou HPS. Esse algoritmo parte do princípio que o espectro da janela de um sinal de áudio é formado por picos de energia situados na fundamental e na série harmônica da nota. O processo seguido pelo algoritmo é ilustrado na Figura 7 e consiste em multiplicar o espectro do sinal, obtido a partir do algoritmo da Transformada, por versões comprimidas do próprio espectro. A compressão é realizada reduzindo a amostragem do espectro original por fatores inteiros.

Figura 7 – Processo da Técnica HPS

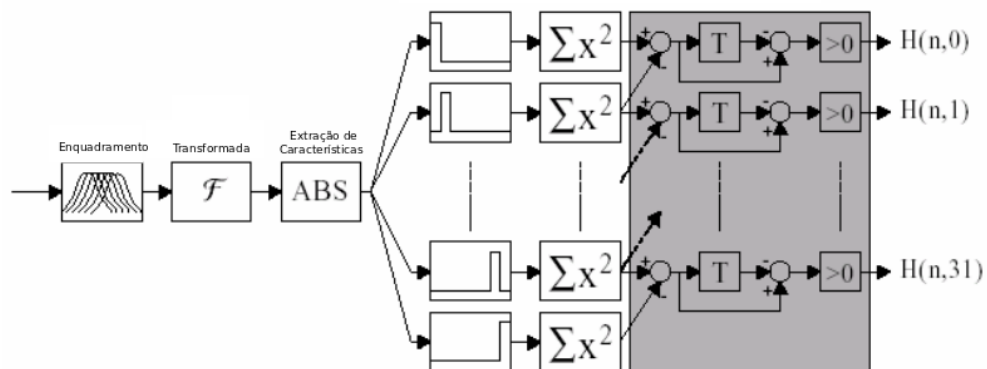


Fonte: (SANTOS, 2011)

Na extração de características, os dados obtidos pela transformada são novamente transformados com o objetivo de reduzir a dimensionalidade e, ao mesmo tempo, aumentar a invariância às distorções do som, para gerar os vetores acústicos finais.

A maioria dos recursos descritos até agora são medições absolutas e, para melhor caracterizar variações temporais no sinal, no pós-processamento, derivadas de tempo de

Figura 8 – Processo de extração de características

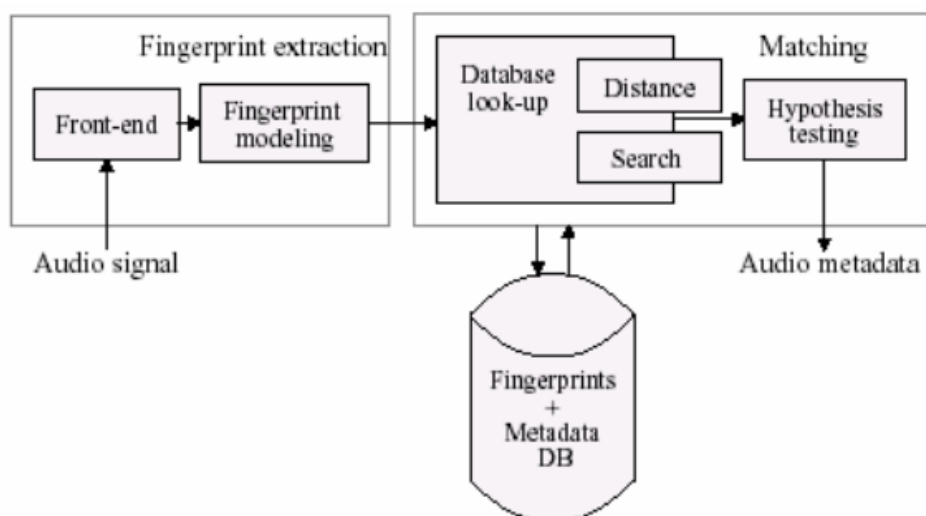


Fonte: (HAITSMA; KALKER, 2002), adaptado pela autora

ordem mais alta são adicionadas ao modelo de sinal (CANO et al., 2005), conforme mostrado na Figura 8.

A modelagem de *fingerprint* geralmente recebe uma sequência de vetores de características calculadas quadro a quadro. As características escolhidas para montar o modelo de *fingerprint* influenciam a forma como o algoritmo irá tratar futuramente a busca dessas características para comparação. Os modelos podem ser consultados em (CANO et al., 2005) e (HAITSMA; KALKER, 2002). Uma forma compacta de *fingerprint* é obtida, e assim, juntamente com informações sobre o formato de áudio original, é enviada para um servidor para identificação, como pode ser visto na Figura 9.

Figura 9 – Modelagem da identificação de áudio



Fonte: (HENRIQUES, 2003)

No segundo processo, denominado *match*, há uma grande diversidade de algoritmos

para localização, identificação e comparação de *fingerprints*. Conforme mencionado anteriormente, o modelo de *fingerprint* escolhido influencia a forma como o algoritmo irá à busca dessas características para comparação. Encontrar *fingerprints* extraídas em um banco de dados de *fingerprints* não é uma tarefa trivial. Em vez de procurar uma *fingerprint* exata (fácil!), a *fingerprint* mais semelhante precisa ser encontrada (HAITSMA; KALKER, 2002).

A questão fundamental é como fazer a comparação eficiente do áudio desconhecido contra os possíveis milhões de *fingerprints*. Uma abordagem de força bruta que calcule as semelhanças entre a *fingerprint* da gravação desconhecida e as armazenadas no banco de dados pode ser proibitiva. O tempo para encontrar uma melhor correspondência é proporcional a  $Nc(d()) + E$ , onde  $N$  é o número de *fingerprints* no repositório e  $c(d())$  o tempo necessário para uma única semelhança, sendo  $E$  algum tempo extra da CPU (CANO et al., 2005).

Alguns algoritmos citados por Cano et al. (2005) são utilizados para medir a similaridade do conteúdo contido no modelo de *fingerprint*:

- *K-Nearest Neighbors* ou K-NN, classificação do vizinho mais próximo usando uma estimativa de entropia cruzada.
- Distância Euclidiana, ou versões modificadas que lidam com sequências de diferentes comprimentos.
- Distância Manhattan, onde as sequências do vetor são quantizadas, ou distância Hamming quando a quantização é binária.

Esses algoritmos são descritos em mais detalhes nas subseções 2.5.1.2 e 2.5.1.7.

#### 2.5.1.2 Classificação

Classificação é a tarefa de aprender uma função alvo  $f$  que mapeie cada conjunto de atributos  $\mathbf{x}$  para um dos rótulos de classes  $\mathbf{y}$  pré-determinados (TAN; STEINBACH; KUMAR, 2009). A função alvo também é conhecida como modelo de classificação e, segundo Tan, Steinbach e Kumar (2009), pode ser classificada de duas formas:

- Modelagem Descritiva: Pode servir como ferramenta explicativa para se distinguir entre objetos e classes diferentes. Por exemplo, explicar quais características definem uma música como Pop, Rock ou Jazz;

- **Modelagem Preditiva:** Pode ser usada para prever o rótulo da classe de registros não conhecidos, sendo tratada como uma caixa preta que atribui automaticamente um rótulo de classe quando recebe o conjunto de atributos de um registro desconhecido.

Técnicas de classificação são mais apropriadas para prever ou descrever conjuntos de dados com categorias nominais ou binárias, sendo menos efetivas para categorias ordinais.

Uma técnica de classificação é uma abordagem sistemática para construção de modelos de classificação a partir de conjuntos de dados de entrada. Cada técnica emprega um algoritmo de aprendizagem para identificar o modelo que seja mais apropriado para o relacionamento entre o conjunto de atributos e o rótulo de classe dos dados de entrada (Ver Algoritmo 1). O modelo gerado deve se adaptar bem aos dados de entrada e prever corretamente os rótulos de classes de registros que ele nunca viu antes (TAN; STEINBACH; KUMAR, 2009).

<p><b>Algoritmo 1:</b> Algoritmo de Classificação básico</p>
--

- |  |
|--|
| <ol style="list-style-type: none"> <li>1 Receber uma coleção de registros;</li> <li>2 Encontrar um modelo para determinar o valor do atributo classe em função dos valores de outros atributos;</li> <li>3 Definir a classe de novos registros.</li> </ol> |
|--|

Os métodos de classificação podem ser divididos de duas formas: classificadores *eager* (espertos) e classificadores *lazy* (preguiçosos). Os classificadores espertos constroem um modelo de classificação capaz de classificar novos registros. Uma vez pronto o modelo, o conjunto de treinamento não é mais utilizado na classificação de novos registros. Exemplos de métodos que se enquadram nesta categoria são Árvores de Decisão, Redes Neurais, Redes Bayesianas e Naïve Bayes, Máquinas de Vetores de Suporte e Regras de Decisão.

Árvores de decisão são os classificadores espertos mais populares. Elas são representações gráficas que consistem em: (i) nodos que representam os rótulos das características (atributo); (ii) arcos que correspondem ao valor de uma característica e; (iii) nodos folha que representam uma classificação. Os métodos baseados em árvores, como o algoritmo ID3 (Ver Algoritmo 2), dividem o espaço de entrada em regiões disjuntas para construir uma fronteira de decisão. As regiões são escolhidas baseadas em uma otimização heurística onde a cada passo os algoritmos selecionam a variável que provê a melhor separação de classes de acordo com alguma função alvo (BOGORNÝ, 2018a).

No caso dos classificadores preguiçosos, cada novo registro é comparado com todo o conjunto de treinamento e é classificado segundo a classe do registro que é mais similar. Um exemplo de método bastante conhecido nesta categoria é o *K-Nearest Neighbors*.

**Algoritmo 2:** Algoritmo ID3 básico

- 1 Selecionar um atributo como sendo o nodo raiz;
- 2 Arcos são criados para todos os diferentes valores do atributo selecionado no passo 1;
- 3 Se todos os exemplos de treinamento (registros) sobre uma folha pertencerem a uma mesma classe, esta folha recebe o nome da classe. Se todas as folhas possuem uma classe, o algoritmo termina;
- 4 Senão, o nodo é determinado com um atributo que não ocorra no trajeto da raiz e arcos são criados para todos os valores. O algoritmo retorna ao passo 3.

O *K-Nearest Neighbors*, ou K-NN, é um classificador onde o aprendizado é baseado na analogia. O conjunto de treinamento é formado por vetores n-dimensionais e cada elemento deste conjunto representa um ponto no espaço n-dimensional (ver Figura 10).

Para determinar a classe de um elemento que não pertença ao conjunto de treinamento, o classificador procura K elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, ou seja, que tenham a menor distância. Estes K elementos são chamados de *K-Nearest Neighbors* (K-vizinhos mais próximos). Verifica-se quais são as classes desses K vizinhos e a classe mais frequente será atribuída à classe do elemento desconhecido. O número de K-vizinhos é um parâmetro livre controlado pelo usuário com o objetivo de obter uma melhor classificação (ver Algoritmo 3).

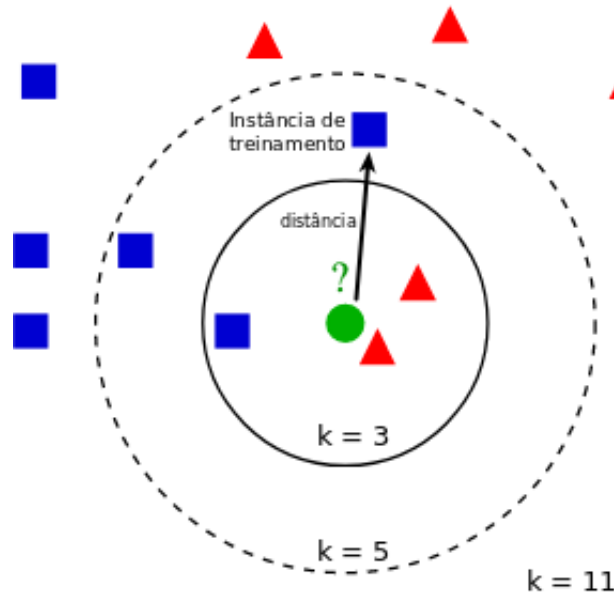
**Algoritmo 3:** Algoritmo K-NN básico

- 1 Selecionar um conjunto de exemplos de treinamento;
- 2 Definir uma métrica para calcular a distância entre os exemplos de treinamento;
- 3 Definir o valor de K (o número de vizinhos mais próximos que serão considerados pelo algoritmo);
- 4 **repita**
  - 5 | Calcular a distância entre o exemplo desconhecido e o outros exemplos do conjunto de treinamento;
- 6 **até que identifique os K vizinhos mais próximos;**
- 7 Utilizar o rótulo da classe dos vizinhos mais próximos para determinar o rótulo de classe do exemplo desconhecido (votação majoritária);

As métricas mais comuns utilizadas para o cálculo da distância são: Distância Euclidiana, Distância de Manhattan e Distância de Minkowski. Este processo de classificação pode ser computacionalmente exaustivo se considerado um conjunto com muitos dados, mas para determinadas aplicações, no entanto, o processo é bem aceitável (SILVA, 2005).



Figura 10 – K-Nearest Neighbors



Fonte: Internet, adaptado pela autora

### 2.5.1.3 Clustering

A formação de agrupamentos (ou *clustering*) divide os dados em grupos (*clusters*) que tenham significado, sejam úteis, ou ambas as coisas. Se *clusters* com significados forem o objetivo, então os *clusters* devem capturar a estrutura natural dos dados.

O objetivo é que os objetos dentro de um *cluster* sejam semelhantes (ou relacionados) entre si e diferentes de (ou não relacionados aos) objetos de outros *clusters*. Quanto maior a semelhança (ou homogeneidade) dentro de um *cluster* e maior a diferença entre *clusters*, melhor ou mais distinto será o agrupamento (TAN; STEINBACH; KUMAR, 2009).

Encontrar o melhor agrupamento para um conjunto de objetos não é uma tarefa simples, a não ser que  $n$  (número de objetos) e  $k$  (número de *clusters*) sejam extremamente pequenos, visto que o número de partições distintas em que podemos dividir  $n$  objetos em  $k$  *clusters* é aproximadamente  $k^n/k!$ , por exemplo, se  $k = 2$  e  $n = 5$  então são 16 formas de dividir 5 elementos em 2 *clusters*.

A técnica de *clustering* está relacionada a outras técnicas que são usadas para dividir objetos de dados em *clusters*. Por exemplo, o *clustering* pode ser considerado uma forma de classificação pelo fato de criar uma rotulagem de objetos com rótulos de classes (*clusters*). Entretanto, ela deriva estes rótulos apenas dos dados (TAN; STEINBACH; KUMAR, 2009).

As principais abordagens de *clustering* são as seguintes:

- **Particionamento (*k-means* ou k-médias):** é uma divisão do conjunto de objetos de dados, em um espaço  $n$ -dimensional contínuo, em subconjuntos (*clusters*) não interseccionados de modo que cada objeto de dado esteja exatamente em um subconjunto, ou seja, divide os dados em conjuntos disjuntos (*clusters*) tal que cada ponto (objeto) pertence a um único cluster (ver Figura 12). É uma técnica particional de agrupamento baseada em protótipos que tenta encontrar um número especificado de *clusters* ( $K$ ) pelo usuário, que são representados pelos seus centroides (ponto médio de um *cluster* de pontos) (TAN; STEINBACH; KUMAR, 2009). A técnica de K-means é simples: Primeiro escolhe-se  $K$  centróides iniciais e o número de *clusters* desejado, que são parâmetros especificados pelo usuário. Cada ponto é atribuído a seguir ao centróide mais próximo, e cada coleção de pontos atribuídos a um centróide é um *cluster*. O centróide de cada *cluster* é então recalculado baseado nos pontos atribuído ao *cluster*. Repete-se os passos de atribuição e recalculo até que os centróides permaneçam os mesmos. (Ver Algoritmo 4 e aplicação do algoritmo na Figura 11). O centróide é recalculado utilizando as medidas de similaridade e distância que podem ser vistas na subseção 2.5.1.7.

**Algoritmo 4:** Algoritmo K-means básico

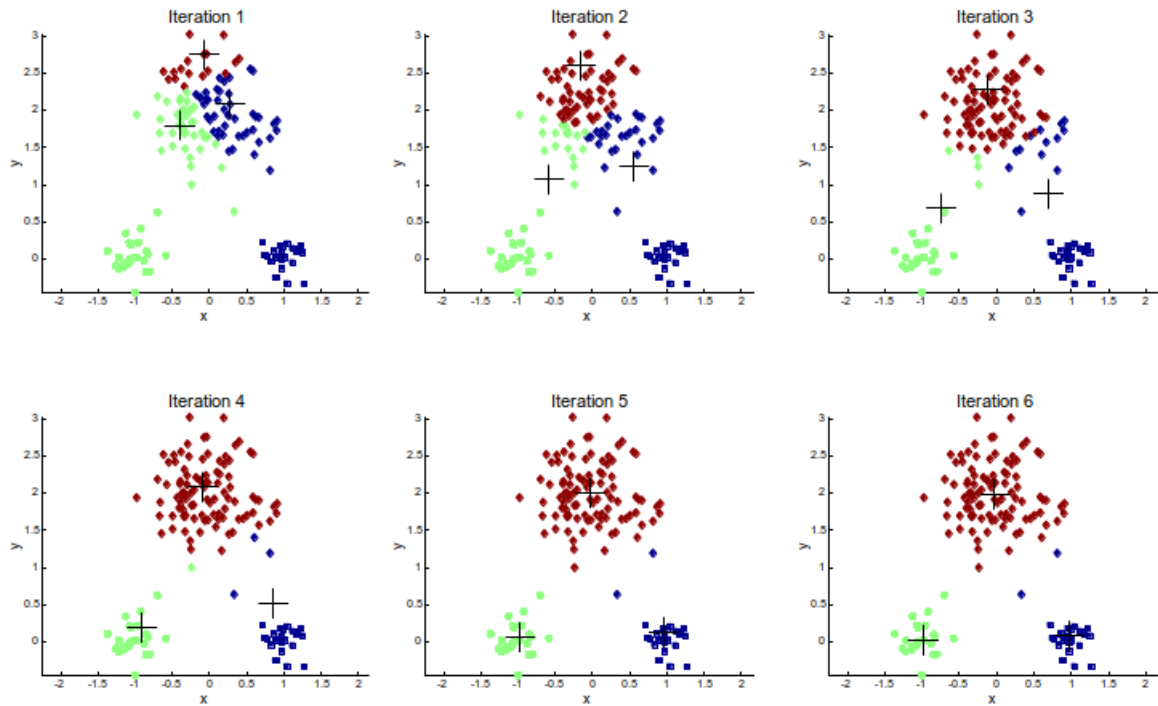
- |   |   |
|---|---|
| 1 | Selecione $K$ pontos como centróides iniciais;                          |
| 2 | <b>repita</b>   |
| 3 | Forme $K$ clusters atribuindo cada ponto ao seu centróide mais próximo; |
| 4 | Recalcule o centróide de cada cluster;                                  |
| 5 | <b>até</b> que os centróides não mudem;                                 |

- **Hierárquico:** se refere a um conjunto de técnicas de *clustering* intimamente relacionadas que produzem um conjunto de *clusters* aninhados organizados como uma árvore. Segundo Tan, Steinbach e Kumar (2009), há duas abordagens básicas para gerar um *cluster* hierárquico:

- **Aglomerativa:** inicia com pontos individuais e, em cada etapa, funde os pares mais próximos em *clusters*. Esta abordagem é expressa no Algoritmo 5;
- **Divisiva:** inicia com um *cluster* contendo todos os pontos e, a cada etapa, divide o *cluster* até que restem apenas *clusters* únicos de pontos individuais. Neste caso, precisamos decidir qual *cluster* dividir em cada etapa e como fazer a divisão. Esta abordagem é expressa no Algoritmo 6;

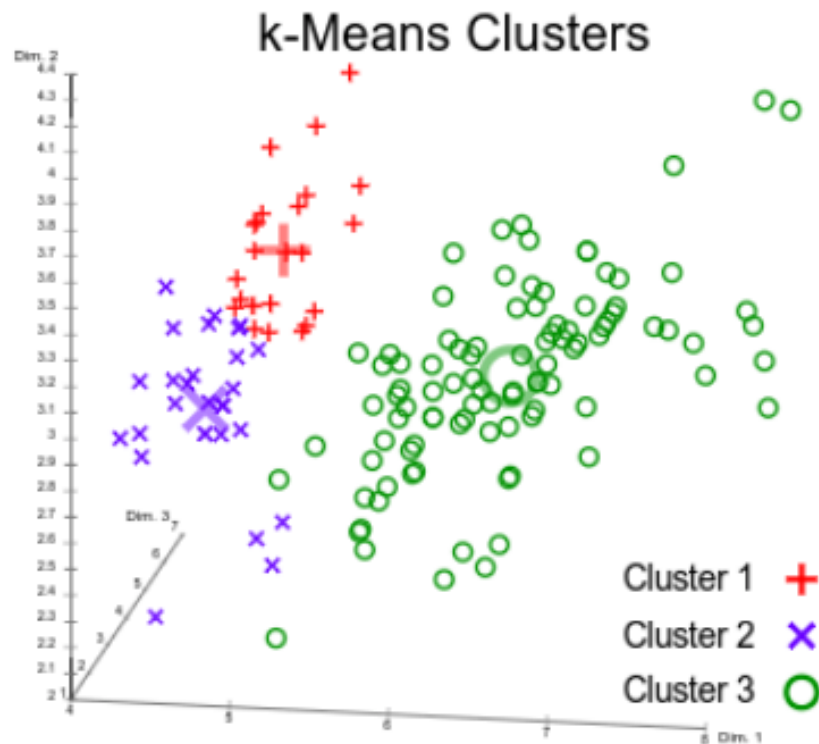
Técnicas de agrupamento hierárquico aglomerativo são as de uso mais comum. A matriz de proximidade é o cálculo da similaridade entre dois *clusters*. Ela é definida

Figura 11 – Usando o algoritmo K-means para encontrar três grupos nos dados de exemplo



Fonte: (BOGORNÝ, 2018b)

Figura 12 – Conjunto de objetos de dados em clusters



Fonte: (ENESCO; DHANDHANIA; SVETLECIC, 2018?)

**Algoritmo 5:** Algoritmo de agrupamento hierárquico aglomerativo básico

- 1 Calcule a matriz de proximidade, caso necessário;
- 2 **repita**
- 3     Fundam os dois *clusters* mais próximos;
- 4     Atualize a matriz de proximidade para refletir a proximidade entre o novo *cluster* e os *clusters* originais;
- 5 **até** que reste apenas um *cluster*;

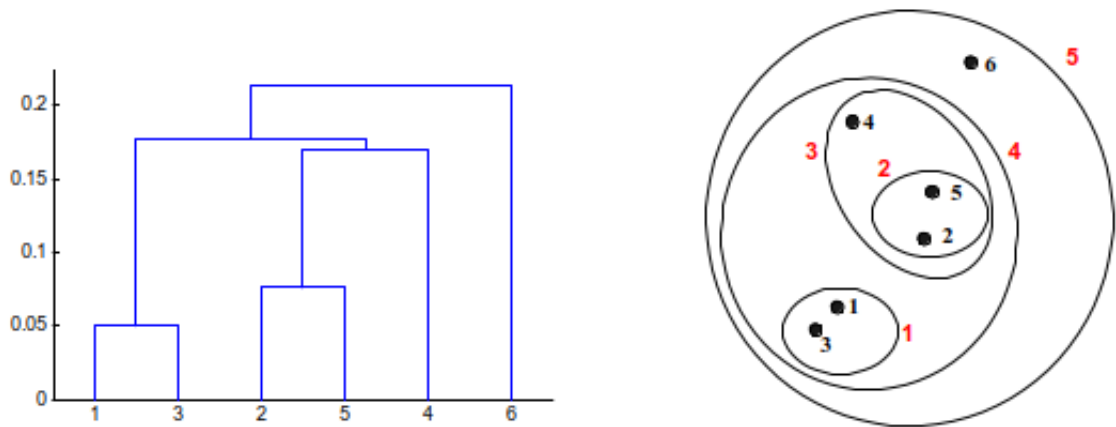
**Algoritmo 6:** Algoritmo de agrupamento hierárquico divisivo básico

- 1 Calcule a matriz de distância, caso necessário;
- 2 **repita**
- 3     Crie um novo *cluster* para dividir o *cluster* correspondente à maior diferença;
- 4 **até** que restem apenas *clusters* únicos;

com um tipo específico de técnica de agrupamento hierárquico, como MIN, MAX e Média do grupo. MIN define a similaridade entre dois *clusters* baseando-se nos dois pontos mais similares (mais próximos) dos dois clusters diferentes. De forma alternativa, MAX define a similaridade entre dois *clusters* baseando-se nos dois pontos menos similares (mais distantes) entre dois *clusters*. Já a Média do Grupo é dada pela média da distância entre pares de pontos nos dois clusters (TAN; STEINBACH; KUMAR, 2009). Uma técnica alternativa, o método de *Ward*, supõe que um *cluster* seja representado pelo seu centróide, mas mede a similaridade entre dois *clusters* em termos do aumento do erro quadrado (similar à média do grupo, porém a distância entre os pontos é a distância ao quadrado) que resulta da fusão dos dois *clusters*. Um cluster hierárquico é exibido frequentemente usando um diagrama do tipo árvore chamado dendograma, que exibe tanto os relacionamentos *cluster*-sub*cluster* quanto a ordem na qual os *clusters* são fundidos ou divididos (TAN; STEINBACH; KUMAR, 2009) (ver exemplo na Figura 13).

- Baseadas em Densidade (DBSCAN): localizam regiões de alta densidade que estejam separadas entre si por regiões de baixa densidade (ESTER et al., 1996), sendo o algoritmo DBSCAN o principal representante desta categoria (TAN; STEINBACH; KUMAR, 2009). A densidade baseada em centro, seguida pelo DBSCAN, é avaliada para um determinado ponto no conjunto de dados contando-se o número de pontos dentro de um determinado raio (*Eps*) a partir daquele ponto, que inclui ele próprio. Desta forma, é possível classificar um ponto como estando: (i) no interior de uma região densa; (ii) no limite de uma região densa; ou (iii) em uma região ocupada esparsamente. Uma região densa é uma região onde cada ponto tem *muitos* pontos

Figura 13 – Um cluster hierárquico de quatro pontos mostrado como um dendograma e como grupos aninhados



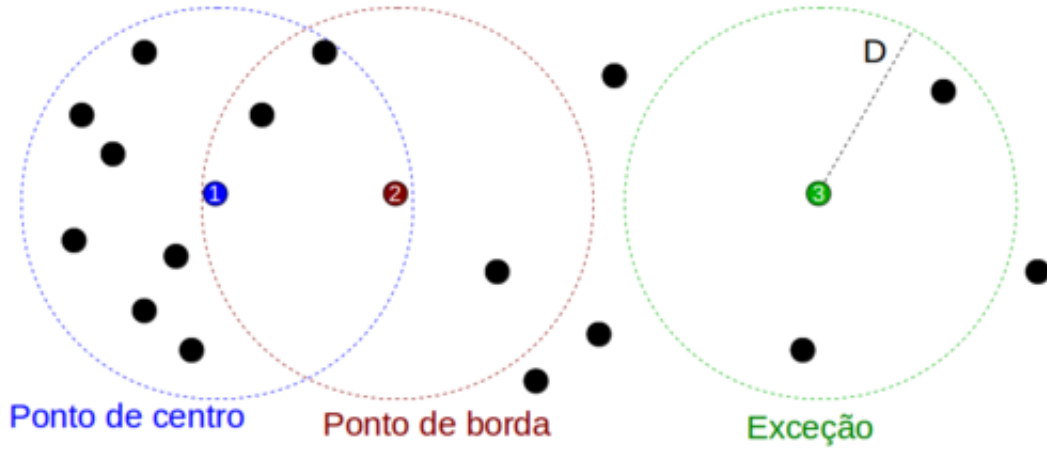
Fonte: (BOGORNÝ, 2018c)

em sua *vizinhança* (BOGORNÝ, 2018c). A vizinhança é determinada pelo raio Eps. Um ponto é central se o número de pontos dentro de uma determinada vizinhança têm pelo menos *MinPts* (parâmetro especificado pelo usuário) em torno do ponto em questão. Um ponto de limite não é um ponto central, mas fica dentro da vizinhança de um ponto central. Um ponto de limite pode cair dentro das vizinhanças de diversos pontos centrais. Já um ponto ruído é qualquer ponto que não seja nem um ponto central nem um de limite (TAN; STEINBACH; KUMAR, 2009). Na Figura 14, o ponto 1 é um ponto central para o raio D (Eps) se  $MinPts = 9$ . O ponto 2 é um ponto de limite e o ponto 3 é um ponto de ruído. O Algoritmo 7 apresenta o DBSCAN. Quaisquer dois pontos de centro que estejam suficientemente próximos - dentro de uma distância Eps entre si - são colocados no mesmo *cluster*. Da mesma forma, qualquer ponto de limite que esteja suficientemente próximo de um ponto de centro é colocado no mesmo *cluster* do ponto de centro. Pontos de ruídos são descartados (TAN; STEINBACH; KUMAR, 2009). O DBSCAN pode lidar com *clusters* de tamanhos e formas arbitrárias, conseguindo encontrar muitos *clusters* que não podem ser encontrados utilizando o K-means.

**Algoritmo 7:** Algoritmo DBSCAN

- 1 Rotular todos os pontos como de centro, de limite ou ruído;
- 2 Eliminar os pontos de ruído;
- 3 Colocar uma aresta entre todos os pontos de centro que estejam dentro da Eps uns dos outros;
- 4 Tornar cada *cluster* de pontos de centro conectados a um *cluster* separado;
- 5 Atribuir cada ponto de limite a um dos *clusters* dos seus pontos de centro associados.

Figura 14 – Eps, pontos de centro, de limite e de ruído



Fonte: (WIKILIVROS, 2018?)

#### 2.5.1.4 Cross Recurrence Plot

O trabalho de Serra e Andrzejak (2009) propõe o uso de *Cross Recurrence Plots* (CRP) como métrica de similaridade entre músicas. Antes de definir CRP, é necessário introduzir o conceito de *Recurrence Plot* (RP). RP é uma ferramenta utilizada para visualizar recorrências com uma série temporal, ou seja, regiões onde a órbita da série passa perto de um estado previamente visitado. Mais especificamente, RP é uma matriz quadrada, preenchida com zeros e uns, que indica se há ou não recorrência, ou seja, se o estado no tempo  $i$  é similar ao estado do tempo  $j$  (ECKMANN; KAMPHORST; RUELLE, 1987; ALLIGOOD; SAUER; YORKE., 1996). A diagonal principal de um RP é, portanto, composta por uns. CRPs são construídos da mesma maneira que RPs, mas cada eixo corresponde a uma série temporal diferente e a matriz resultante não é quadrada.

Primeiramente, o algoritmo extrai a característica HPCP (GUTIÉRREZ, 2006) de duas músicas, resultando em séries temporais de  $H = 12$  variáveis. Dados os vetores HPCP da música  $\mathbf{x}$  e da música  $\mathbf{y}$ , calcula-se a transposição de  $\mathbf{y}$  de modo que ela fique na mesma tonalidade de  $\mathbf{x}$ . A transposição ocorre rotacionando-se o vetor HPCP de  $\mathbf{y}$  em  $k$  posições por meio da técnica *Optimal Transposition Index*, proposta em (SERRA; ANDRZEJAK, 2009).

A seguir, calcula-se o *embedding* das duas músicas em um espaço de fase, isto é, um espaço onde as recorrências do sinal podem ser obtidas. Considere que HPCP  $\mathbf{x}$  tem  $N_x^*$  janelas. O *embedding* de  $\mathbf{x}$  é dado por  $\mathbf{x}' = \{x_i\}$ , para  $i = 1, \dots, N_x$ ,  $N_x = N_x^* - (m - 1) * \tau$ ,

em que  $x_i$  é calculado conforme segue:

$$\begin{aligned}
 x_i = & \quad (x_{1,i}, x_{1,i+\tau}, \dots, x_{1,i+(m-1)\tau}, \\
 & \quad x_{2,i}, x_{2,i+\tau}, \dots, x_{2,i+(m-1)\tau}, \\
 & \quad \vdots \\
 & \quad x_{H,i}, x_{H,i+\tau}, \dots, x_{H,i+(m-1)\tau})
 \end{aligned} \tag{2.1}$$

Os autores estimaram os valores ótimos de  $m$  e  $\tau$  para o reconhecimento de músicas *cover* por meio da divisão de uma base de dados em conjunto de treinamento e teste. Os parâmetros encontrados foram  $m = 10$  e  $\tau = 1$ .

Serra e Andrzejak (2009) utilizam a Equação 2.2 para calcular o CRP, em que  $\Theta(\cdot)$  é a função degrau tipo Heaviside ( $\Theta(v) = 0$  se  $v < 0$  e  $\Theta(v) = 1$ , caso contrário),  $\epsilon_i^x$  e  $\epsilon_i^y$  são limiares de distâncias e  $\|\cdot\|$  é a norma Euclidiana. No artigo, os autores calculam os limiares dinamicamente, de modo que 10% dos vizinhos de cada entrada sejam considerados semelhantes.

$$R_{i,j} = \Theta(\epsilon_i^x - \|\mathbf{x}_i - \mathbf{y}_j\|) \Theta(\epsilon_i^y - \|\mathbf{x}_i - \mathbf{y}_j\|) \tag{2.2}$$

Em geral, pares de músicas diferentes não exibem nenhum padrão evidente no CRP e pares de músicas *cover* apresentam estrutura de linhas longas e, caso contrário, tais padrões não ocorrem.

Com base no CRP de duas músicas, o trabalho de Serra e Andrzejak (2009) propõe o uso do maior comprimento das diagonais formadas na matriz como métrica de similaridade. A medida  $Q_{max}$  é definida como o maior comprimento das diagonais na matriz CRP, considerando possíveis variações no tempo da música (que correspondem a curvaturas nos traços) e na melodia (pequenas rupturas). Uma análise detalhada para o cálculo do  $Q_{max}$  pode ser encontrada em (SERRA; ANDRZEJAK, 2009).

A partir da medida  $Q_{max}$ , é possível identificar grupos de músicas *cover* em uma base de dados. Quanto maior o  $Q_{max}$  entre duas instâncias, maior a probabilidade de elas serem interpretações diferentes da mesma música. Até hoje,  $Q_{max}$  é considerado o estado da arte em métricas para identificação de músicas *cover*. Sua principal desvantagem é seu alto custo computacional.

#### 2.5.1.5 Dynamic Time Warping

O *Dynamic Time Warping* (DTW) (KEOGH; RATANAMAHATANA, 2004) é uma técnica que permite definir uma métrica entre séries temporais, como por exemplo, caracte-

rísticas de áudio. Dadas duas séries  $Q$  e  $C$  de tamanhos  $n$  e  $m$ , respectivamente, cria-se uma matriz  $D$ ,  $n \times m$ , onde o elemento  $d_{i,j}$  é a distância entre o elemento  $q_i$  e  $c_j$ .

A partir da matriz  $D$ , o algoritmo procura por um caminho mínimo  $W$ , onde  $w_k = (i, j)_k$ , que respeite as seguintes condições:

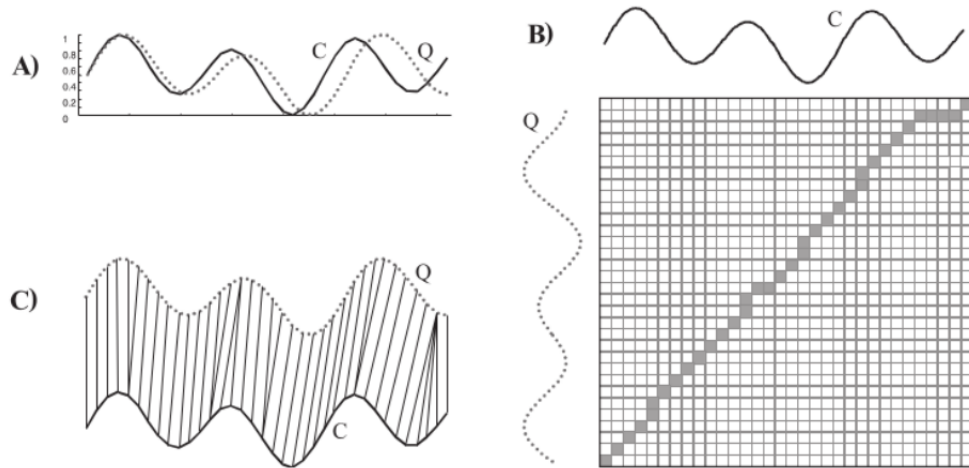
- Comece em  $(1, 1)$  e termine em  $(n, m)$ ;
- Percorra apenas índices adjacentes;
- Percorra espaçamentos iguais no tempo.

O caminho mínimo é encontrado por meio de um algoritmo de programação dinâmica de complexidade  $O(nm)$  (KEOGH; RATANAMAHATANA, 2004). A métrica de dissimilaridade DTW é dada por:

$$DTW(Q, C) = \min \left\{ \sqrt{\sum_{k=1}^K D[w_k]} \right\} \quad (2.3)$$

em que  $K$  é o tamanho do caminho e  $D[w_k]$  é o valor da entrada  $(i, j)_k$  na matriz  $D$ . A Figura 15 ilustra a execução do algoritmo DTW.

Figura 15 – Execução do algoritmo de DTW. (A) Séries  $C$  e  $Q$ . (B) Matriz de distâncias. (C) Alinhamento das duas séries com o caminho mínimo.



Fonte: (KEOGH; RATANAMAHATANA, 2004)

Seu uso é inviável em grandes conjuntos de dados. Por este motivo, diversas técnicas de indexação específicas para esse algoritmo foram propostas para a tarefa de busca por similaridade, tais como *lower bound* e *early abandoning*. Com essas técnicas, é possível encontrar os vizinhos mais próximos de uma dada subsequência em uma quantidade massiva de séries temporais. Informações mais detalhadas podem ser consultadas em (MIZUTANI, 2006), (KRUSKAL; LIBERMAN, 1983) e (JUANG; RABINER, 1991).



### 2.5.1.6 Indexação

Estruturas de indexação (uma descrição sobre essas estruturas pode ser encontrada em (GARCIA-MOLINA; ULLMAN; WIDOM, 2002)) são normalmente fornecidas pelos SGBDs. A idéia básica dessas estruturas consiste na escolha de um objeto arbitrário central e na aplicação de uma função de distância para dividir os demais objetos em vários subconjuntos. Dessa maneira, uma estrutura de indexação é construída executando-se esse mesmo procedimento, recursivamente, para cada subconjunto não vazio (BARIONI, 2006).

Por exemplo, a *Slim-tree* (um método aplicado à indexação de dados musicais) é uma árvore dinâmica e balanceada que cresce a partir das folhas em direção a raiz. Ela agrupa os objetos de um conjunto de dados em páginas de tamanho fixo, sendo que cada página corresponde a um nó da árvore. A *Slim-tree* armazena todos os objetos nas folhas, organizando-os hierarquicamente na árvore. Essa hierarquia é construída a partir da seleção de objetos, denominados objetos representantes, que definem centros de regiões no espaço de dados. Cada região possui um raio de cobertura, e apenas os objetos que forem cobertos pelo raio de cobertura de uma determinada região podem ser armazenados nesse nó. Desta forma, cada nó da árvore (exceto o nó raiz) possui, basicamente, um objeto representante, um raio de cobertura e os objetos do conjunto de dados que estão cobertos pela região do nó (TRAINA et al., 2000).

Assim como a *Slim-tree*, na literatura podem ser encontradas outros métodos aplicados à indexação de dados musicais, são a *VP-tree* (*Vantage Point tree*) (YIANILOS, 1993), a *MVP-tree* (*Multi-Vantage Point tree*) (BOZKAYA; ÖZSOYOGLU, 1997), a GNAT (*Geometric Near-neighbor Access Tree*) (BRIN, 1995), a *M-tree* (CIACCIA; PATELLA; ZEZULA, 1997), *Família-Omni* e a *DBM-tree* (FILHO et al., 2001; VIEIRA; CHINO; TRAINA, 2004), a *n-grams* (DOWNIE, 1999) e o *Vantage Indexing Method* (TYPKE et al., 2003).

### 2.5.1.7 Medidas de Similaridade

A semelhança entre dois objetos é uma medida numérica do grau no qual os dois objetos se parecem. A diferença, ou dissimilaridade, entre dois objetos é uma medida numérica do grau no qual os dois objetos são diferentes. Frequentemente, o termo distância é usado como sinônimo de diferença, embora, muitas vezes seja usado para se referir a uma classe especial de diferenças.

Similaridade e distância são importantes pois são utilizadas por inúmeras técnicas de mineração de dados, como *clustering* (ver subseção 2.5.1.3) e classificação (ver subseção 2.5.1.2). Quanto menor o valor desta distância, mais semelhantes serão os objetos e eles

tenderão a ficar no mesmo cluster. Quanto maior a distância, menos similares serão os objetos e, em consequência, eles deverão estar em grupos distintos.

Uma função de distância deve ser tal que:

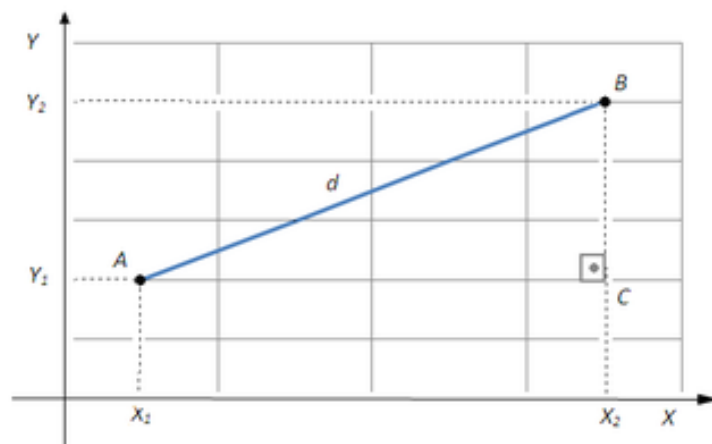
- Não assuma valores negativos (o menor valor é zero);
- Seja simétrica (a distância do objeto  $i$  ao objeto  $j$  deve ser a mesma distância do objeto  $j$  ao  $i$ );
- Forneça o valor zero quando calculada a distância do objeto a si mesmo ou quando dois objetos são idênticos;
- Respeite a desigualdade triangular (dados 3 objetos, a distância entre dois deles tem que ser menor ou igual à soma das distâncias entre esses dois objetos e o terceiro).

Alguns algoritmos na literatura são utilizados para medir distância e similaridade:

- Distância Euclidiana ou Distância Métrica: frequentemente utilizada com uma medida de distância entre dois pontos em planos n-dimensionais (ver Figura 16), que pode ser provada pela aplicação repetida do teorema de Pitágoras. À medida que o número de dimensões aumenta, o tempo de cálculo também aumenta. A distância euclidiana entre os pontos  $P = (p_1, p_2, \dots, p_n)$  e  $Q = (q_1, q_2, \dots, q_n)$  num espaço euclidiano n-dimensional é expressa na Equação 2.4.

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.4)$$

Figura 16 – Distância Euclidiana entre os pontos A e B

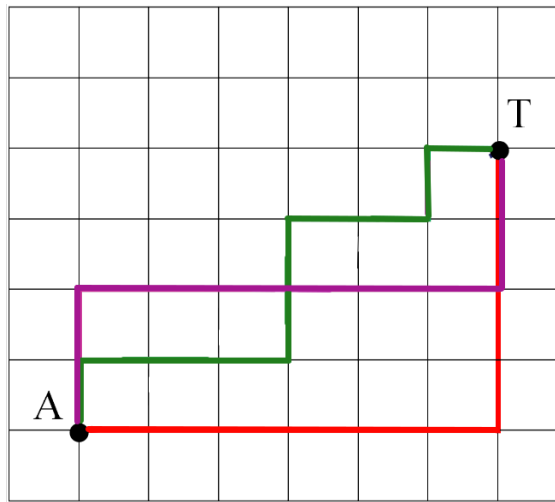


Fonte: Internet

- Distância de Manhattan: medida de distância entre dois pontos em um espaço euclidiano com um sistema de cartesiano de coordenadas fixas (ver Figura 17). É a soma dos comprimentos da projeção da linha que une os pontos com os eixos das coordenadas. Em um plano que contém os pontos  $P_1$  e  $P_2$ , com as coordenadas  $(x_1, y_1)$  e  $(x_2, y_2)$  respectivamente, essa distância é definida através da Equação 2.5.

$$|x_1 - x_2| + |y_1 - y_2| \quad (2.5)$$

Figura 17 – Distância de Manhattan (linhas roxa, verde e vermelha)



Fonte: Internet

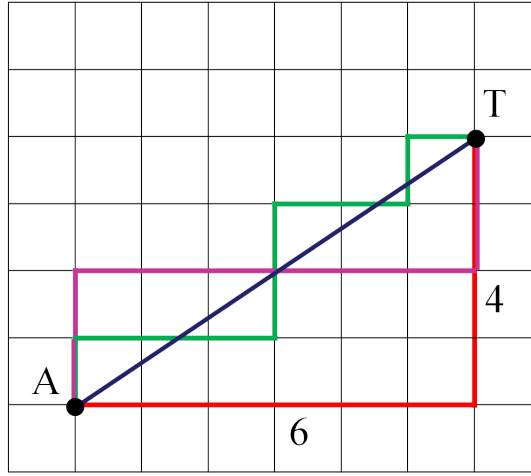
- Distância de Minkowski: generalização das distâncias Euclidiana e de Manhattan (ver Figura 18), sendo definida pela Equação 2.6. Quando  $q = 1$ , esta distância representa a distância de Manhattan e quando  $q = 2$ , a distância Euclidiana.

$$d(x, y) = (|x_1 - y_1|_q + |x_2 - y_2|_q + \dots + |x_n - y_n|_q)^{\frac{1}{q}}, \text{ onde } q \in N \quad (2.6)$$

- *Longest Common Subsequence* (LCS): maior subsequência de caracteres comuns que há entre dois objetos (ver Figura 19). Dadas duas sequências  $X$  e  $Y$  de comprimento  $m$  e  $n$ , respectivamente, onde  $X = \{X_1, X_2, \dots, X_m\}$  e  $Y = \{Y_1, Y_2, \dots, Y_n\}$  define-se  $LCS(X, Y)$  como a máxima subsequência comum entre  $X$  e  $Y$ . A equação de recorrência para o cálculo de  $LCS(X, Y)$  pode ser deduzida a partir de duas propriedades:

1. Se  $X_1 = Y_1$ , então  $LCS(X, Y)$  é a concatenação de  $X_1$  com  $LCS(X_{2:m}, Y_{2:n})$ .
2. Se  $X_1 \neq Y_1$ , então  $LCS(X, Y) = \max(LCS(X_{2:m}, Y), LCS(X, Y_{2:n}))$ .

Figura 18 – Distância de Manhattan (linhas rosa, verde e vermelha) e Distância Euclidiana (linha azul)



Fonte: Internet

Estas propriedades constituem a subestrutura ótima para a resolução continuada do primeiro caractere de cada string com a aplicação recursiva dessas mesmas propriedades ao sufixo remanescente. As mesmas propriedades são válidas quando as strings são analisadas do fim para o começo, resolvendo o último caractere e aplicando a recursão ao prefixo remanescente:

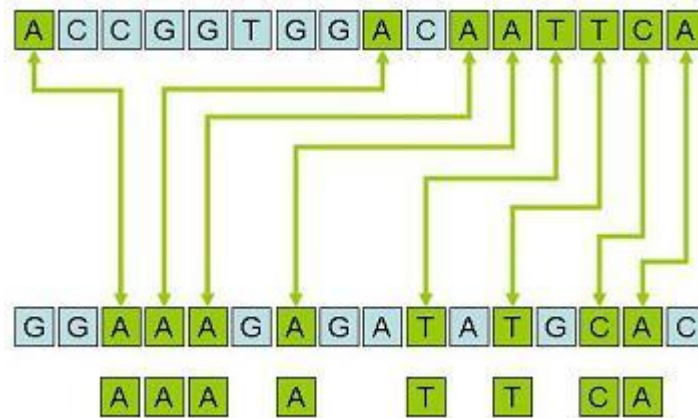
1. Se  $X_m = Y_n$ , então  $LCS(X, Y)$  é a concatenação de  $LCS(X_{1:m-1}, Y_{1:n-1})$  com  $X_m$ .
2. Se  $X_m \neq Y_n$ , então  $LCS(X, Y) = \max(LCS(X_{1:m-1}, Y), LCS(X, Y_{1:n-1}))$ .

As definições matemáticas das recursões por sufixo e prefixo são respectivamente apresentadas nas Equações 2.7 e 2.8.

$$LCS(X, Y) = \begin{cases} \emptyset & \text{se } m = 0 \mid n = 0 \\ X_1 + LCS(X_{2:m}, Y_{2:n}) & \text{se } X_1 = Y_1 \\ \max(LCS(X, Y_{2:n}), LCS(X_{2:m}, Y)) & \text{se } X_1 \neq Y_1 \end{cases} \quad (2.7)$$

$$LCS(X, Y) = \begin{cases} \emptyset & \text{se } m = 0 \mid n = 0 \\ LCS(X_{1:m-1}, Y_{1:n-1}) + X_m & \text{se } X_1 = Y_1 \\ \max(LCS(X_{1:m-1}, Y), LCS(X, Y_{1:n-1})) & \text{se } X_1 \neq Y_1 \end{cases} \quad (2.8)$$

Figura 19 – Longest Common Subsequence



Fonte: Internet

Este algoritmo é apresentado com mais detalhes no trabalho dos autores Cormen et al. (2009).

Não há uma medida de similaridade que sirva para todos os tipos de variáveis que podem existir numa base de dados.

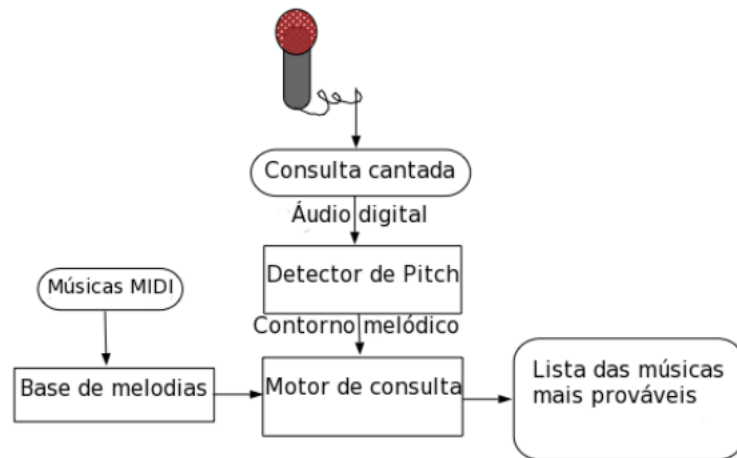
#### 2.5.1.8 Query by Humming

A tarefa de busca musical a partir de um trecho de música cantada ou cantarolada pelo usuário passou a ser conhecida na literatura como *query by humming*, ou QBH. Trata-se de um sistema capaz de reconhecer música pelo *casamento aproximado* de cadeias de caracteres. Ghias et al. (1995), autores do projeto, aplicaram o conceito de *contorno melódico*, ou seja, a forma natural como nós percebemos a música. A Figura 20 mostra a arquitetura do sistema.

Segundo Santos (2011), existem dois principais interesses em QBH: Primeiramente, permitir ao usuário identificar uma música da qual não conheça (ou lembre de) qualquer metadado associado (autor, álbum, título da música, entre outros). Em segundo, é dispor de uma maneira mais natural para consultar coleções de músicas digitais armazenadas principalmente em dispositivos portáteis.

A base de dados de um sistema de QBH é, frequentemente, construída a partir da transposição de músicas MIDI para o formato de representação musical adotado pelo sistema (formatos vistos na seção 2.2). Para casos em que a técnica de reconhecimento baseia-se em propriedades estatísticas associadas à incerteza do canto, é necessário também estimar os parâmetros do modelo. Para estimá-los, no entanto, precisamos de gravações da mesma música por diferentes usuários. Por este motivo, uma plataforma com

Figura 20 – Modelagem do Sistema de QBH



Fonte: (SANTOS, 2011)

este algoritmo estará em constante crescimento e aprendizado.

Capturar o sinal de áudio que codifica uma música cantada por um usuário, seja para construir a base de dados, seja para consultar o sistema por músicas conhecidas, requer atenção quanto ao nível de ruído do ambiente, à taxa de amostragem e à expressão fonética permitida.

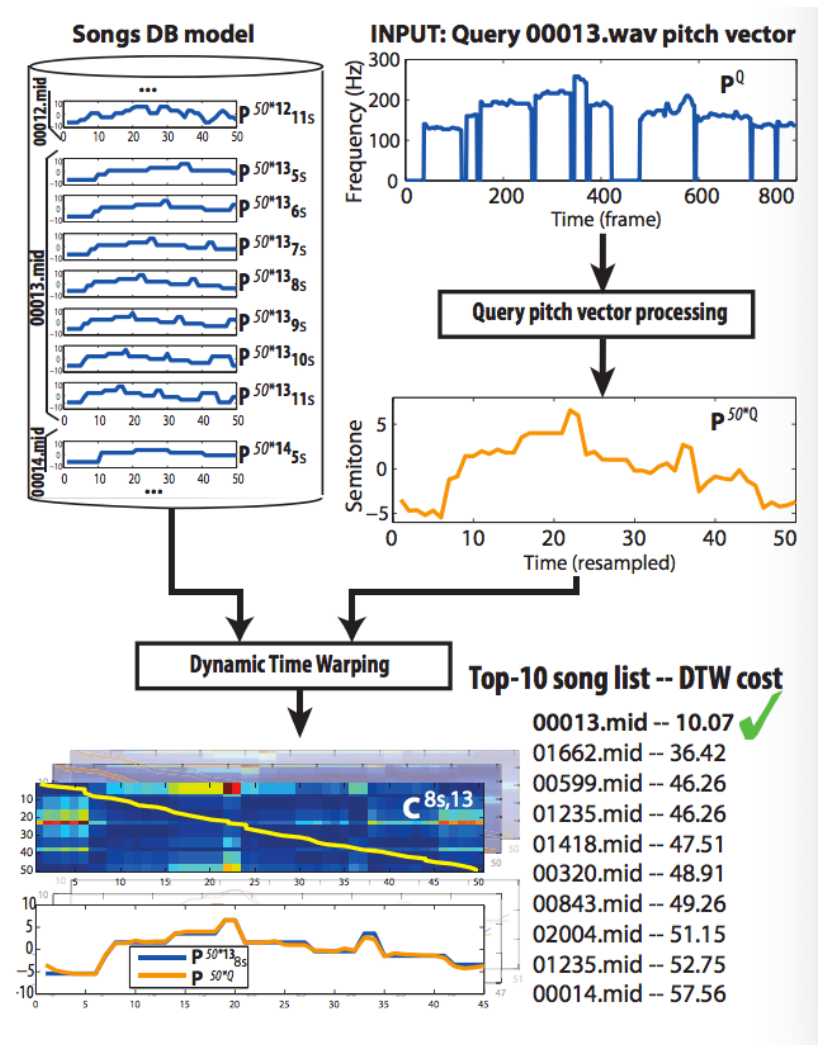
Conforme visto na subseção 2.5.1.1, alguns métodos utilizados no *frontend* para a criação de uma *fingerprint* também são utilizados para a segmentação e enquadramento do som utilizado pelo algoritmo QBH, deixando a representação musical em função do tempo.

Em seguida, a tarefa executada no sistema QBH é propor ao usuário uma lista de músicas que melhor correspondam à consulta, adotando uma medida de dissimilaridade apropriada entre os arquivos. A similaridade entre as músicas no repositório e a consulta é calculada por meio do DTW, visto na subseção 2.5.1.5, que o torna particularmente adequado para recuperação em sistemas QBH. O processo pode ser visto na Figura 21.

Desta forma, vários tipos de sistemas QBH e suas derivações foram pesquisados:

- Wang et al. (2008) propuseram o sistema de QBH combinando as distâncias do *earth mover's distance* (EMD) e do *dynamic time warping* (DTW) baseado na regra de soma ponderada.
- Ryyänen e Klapuri (2008) propuseram o método de extração dos vetores *pitch* através do uso de uma janela de tempo de tamanho fixo e comparando-os por meio do método *locality sensitive hashing* (LSH).
- Salamon e Rohrmeier (2009) propuseram recuperação em dois estágios para o sis-

Figura 21 – Sistema de QBH



Fonte: ACRCLOUD

tema QBH. No primeiro estágio, o número de candidatos é reduzido através do método de indexação utilizando *ngrams*. Depois, um método de comparação mais sofisticado é aplicado com os candidatos restantes baseado no alinhamento local com funções de custos modificados.

- Yu, Tsai e Wang (2006) propuseram métodos para extrair a melodia de vocais das músicas de karaokê, reduzindo a interferência dos acompanhamentos de fundo. Em paralelo, aplicando o *bayesian information criterion* (BIC) para detectar o tempo de início de cada frase no canal vocal acompanhado, o que permite a comparação de similaridade baseada em *dynamic time warping* (DTW) para que seja executada de forma mais eficiente.
- Suzuki et al. (2006) propuseram um método de reconhecimento de letras usando informações de melodia. A informação da melodia é extraída de uma música de

entrada usando informações de alinhamento de tempo, e uma hipótese pode ser verificada do ponto de vista melódico. Um autômato de estado finito (FSA) é usado como gramática de reconhecimento, e uma pontuação total é calculada a partir da pontuação de reconhecimento e da pontuação de verificação.

- Segundo Mesaros (2013), há os métodos e aplicações que lidam com a análise dos sinais de áudio de voz cantada, relacionados à identidade do cantor e ao conteúdo das letras do canto, onde a identificação do cantor em música polifônica é baseada em métodos gerais de classificação de áudio. A classificação de vozes cantadas pode ser feita de maneira robusta em música polifônica quando se usa separação de fontes. A precisão do reconhecimento de palavras da transcrição das letras do canto é bastante baixa, mas mostra-se útil em uma aplicação de consulta por canto, para realizar uma pesquisa textual baseada nas palavras reconhecidas da consulta.

Embora os sistemas atuais de última geração para QBH tenham alcançado um desempenho razoável em casos do mundo real, ainda há muito espaço para a melhoria na indexação de músicas e o método de correspondência do sistema QBH, especialmente em um banco de dados de música na escala da Web.

#### 2.5.1.9 Recuperação por Conteúdo

A recuperação por conteúdo aplica-se a vários tipos de dados, inclusive música, e com o aumento da disponibilização de coleções de áudio digitais, se tornou importante permitir o gerenciamento automático desses tipos de dados por meio da utilização de metadados (ver seção 2.2). Assim, várias técnicas de extração de metadados foram desenvolvidas, sendo algumas delas especialmente adequadas à recuperação de música por conteúdo.

As técnicas baseadas em busca por conteúdo utilizam metadados (como título, álbum e gênero da música) extraídos automaticamente dos dados musicais para representar e indexar as informações embutidas nos áudios digitais. O conjunto de metadados extraídos é chamado de *vetor de características* (TZANETAKIS; ERMOLINSKIY; COOK, 2002).

Nos sistemas de recuperação por conteúdo, as operações de comparação entre dados musicais utilizam os vetores de características para medir a similaridade do conteúdo presente nos dados que eles representam. Segundo Barioni (2006), esse sistema possui quatro componentes principais:

- Um módulo responsável pela extração automática de características que representem o conteúdo presente nos dados complexos;



- Um conjunto definido de métricas capazes de avaliar a similaridade entre os dados musicais;
- Uma interface de usuário que suporte tanto a definição dos parâmetros para a solicitação da consulta aos dados musicais quanto a visualização dos resultados obtidos;
- Um mecanismo de busca, que realiza as operações de busca sobre o conjunto de dados armazenados.

Várias áreas de pesquisa têm contribuído para o desenvolvimento de técnicas relacionadas a um ou mais dos componentes descritos acima. Dentre elas estão as técnicas para a extração automática de características de dados musicais; para a organização, indexação e consulta dos dados musicais; e para o gerenciamento de bases de dados musicais.

O módulo de extração de características é uma das bases fundamentais dos sistemas de recuperação de dados musicais por conteúdo. A sua importância está relacionada ao fato de que são as características extraídas por esse módulo que são utilizadas para a realização da indexação e da recuperação de dados musicais. O processo de extração de características consiste no cálculo de representações numéricas que podem ser utilizadas para caracterizar um determinado dado musical (TRAINA; TRAINA-JR., 2003).

Ao contrário das aplicações tradicionais de bases de dados que manipulam dados numéricos e textuais por meio da realização de consultas por igualdade e ordem, as aplicações que lidam com dados musicais requerem a realização de consultas por similaridade, ou seja, consultas que realizam busca por objetos da base que sejam similares a um objeto de consulta, de acordo com uma certa medida de similaridade. Para que um sistema de recuperação por conteúdo possa responder a consultas por similaridade, é necessário que ele seja capaz de mensurar o quão similar são os diferentes pares de objetos armazenados na base de dados bem como com o objeto de consulta, e é por meio da aplicação de métodos e algoritmos (ver subseção 2.5.1) que é obtida a quantificação dessa similaridade (BOHM; BERCHTOLD; KEIM, 2001; CHÁVEZ et al., 2001).

Porém, essa estratégia não é a mais adequada para ser utilizada em grandes conjuntos de dados, uma vez que o custo computacional envolvido é muito alto. Assim sendo, outro aspecto importante na recuperação de dados musicais por conteúdo está relacionado à utilização de estruturas de indexação (ver subseção 2.5.1.6) apropriadas para espaços métricos que possam agilizar a realização de consultas por similaridade, ou seja, minimizar o número de cálculos de distância necessários para executar uma consulta.

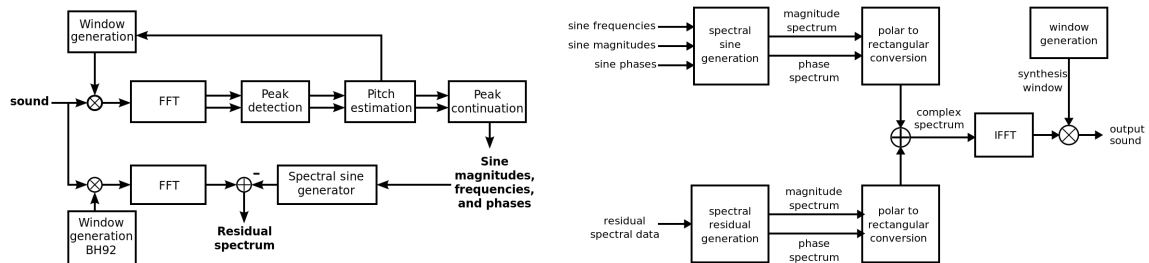
Murthy e Koolagudi (2018) apresentam os aspectos essenciais das tarefas básicas de MIR, como identificação de artista, classificação de gêneros, classificação de emoções mu-

sicais, identificação de instrumentos e anotação musical. Juntamente com essa informação específica da tarefa, os autores também abordam os recursos gerais e os classificadores usados no MIR. As aplicações de cada tarefa MIR para a indústria musical são discutidas em detalhes.

#### 2.5.1.10 Spectral Modeling Synthesis

*Spectral Modeling Synthesis* (SMS), ou na sua tradução, Síntese de Modelagem Espectral, é uma abordagem de modelagem acústica para voz e outros sinais musicais. Ele considera os sons como uma combinação de conteúdo harmônico e conteúdo de ruído. Os componentes harmônicos são identificados com base em picos no espectro de frequência do sinal, normalmente encontrados pela transformada de Fourier de curta duração (FFT). O sinal que permanece após a remoção dos componentes espectrais, por vezes referido como residual, é modelado como ruído branco passado através de um filtro que varia no tempo. A saída do modelo, então, são as frequências e níveis dos componentes harmônicos detectados e os coeficientes do filtro que variam no tempo.

Figura 22 – Análise de SMS e diagramas de bloco de síntese



Fonte: (BONADA et al., 2001)

Na Figura 22, o procedimento de análise primeiro extrai as trajetórias senoidais rastreando os picos em uma sequência de FFTs. Esses picos são então removidos por subtração espectral. O "nível de ruído" restante é modelado como ruído branco por meio de um filtro que varia no tempo. Uma aproximação linear por partes ao envelope espectral superior do ruído é calculada para cada espectro sucessivo, e a parte estocástica é sintetizada por meio da técnica de sobreposição-adição.

Intuitivamente, o modelo pode ser aplicado a muitos tipos de sinais de áudio. Os sinais de fala, por exemplo, incluem a mudança lenta dos sons harmônicos causados pela vibração das cordas vocais e sons de banda larga, semelhantes a ruídos, causados pelos lábios e pela boca. Os instrumentos musicais também produzem sons contendo componentes harmônicos e sons percussivos de ruído quando as notas são tocadas ou alteradas.

O modelo básico e a implementação foram desenvolvidos por Serra e Smith (1990). Desde então, muitas extensões foram propostas, inclusive o SMS Tools, um conjunto de técnicas e implementações de software para análise, transformação e síntese de sons musicais baseados em várias abordagens de modelagem espectral. A técnica SMS provou fornecer transformações gerais de alta qualidade para uma ampla variedade de sinais musicais.

Essas técnicas podem ser usadas para aplicações de síntese, processamento e codificação, enquanto alguns dos resultados intermediários também podem ser aplicados a outros problemas relacionados à música, como separação de fontes sonoras, acústica musical, percepção musical ou análise de desempenho. Informações mais detalhadas sobre a técnica pode ser encontradas em (SERRA; SMITH, 1990).

#### 2.5.1.11 Visualização

Visualização de dados musicais é a exibição na forma de um gráfico ou tabela. Uma visualização bem sucedida requer que os dados sejam convertidos em um formato visual de modo que as características dos mesmos e dos relacionamentos entre itens de dados possam ser analisados ou reportados. O objetivo da visualização é a interpretação da informação visualizada por uma pessoa e a formação de um modelo mental das informações (TAN; STEINBACH; KUMAR, 2009).

A principal motivação para o uso da visualização é que as pessoas podem absorver rapidamente grandes quantidades de informações visuais e encontrar padrões nas mesmas. Há diversas técnicas de visualização mencionadas na literatura. Este trabalho detalha a *Self-Organizing Map* (SOM), em sua tradução, Mapas Auto-Organizados, que é utilizada por uma das soluções acadêmicas analisadas neste trabalho.

A SOM é um tipo de rede neural artificial (ANN) que é treinada usando aprendizagem não supervisionada para produzir uma representação de baixa dimensionalidade (tipicamente bidimensional), chamado de mapa. Assim como em outros tipos de agrupamento baseado em centróides, como o K-means, o objetivo da SOM é encontrar um conjunto de centróides (vetores de referência) e atribuir cada objeto no conjunto de dados ao centróide que fornece a melhor aproximação desse objeto. Uma característica que distingue a SOM de outras abordagens de agrupamento é que ela impõe uma organização topográfica (espacial) pré-determinada sobre os centróides (TAN; STEINBACH; KUMAR, 2009). Em uma descrição de alto nível, a técnica SOM consiste dos passos descritos no Algoritmo 8.

Como derivação da SOM, há a Emergent SOM (ESOM), um mapa topográfico de

**Algoritmo 8:** Algoritmo SOM básico

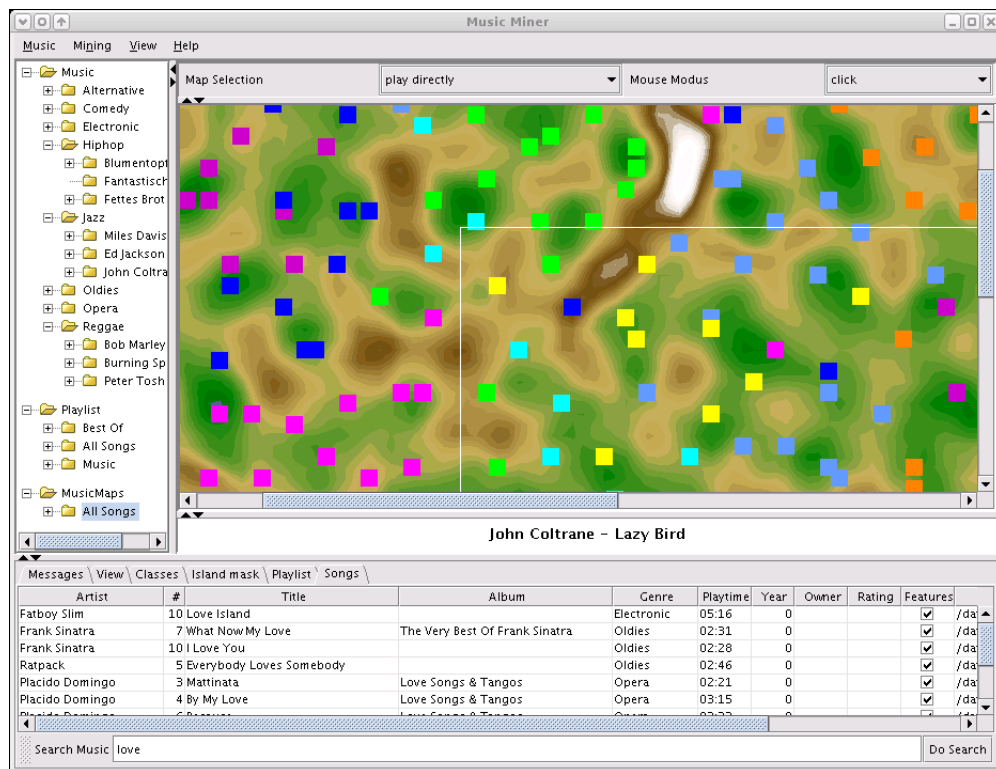
- 1 Inicialize os centróides;
- 2 **repita**
- 3    Selecione o próximo objeto;
- 4    Determine o centróide mais próximo do objeto;
- 5    Atualize este centróide e os centróides que estiverem próximos, i.e., em uma vizinhança especificada;
- 6 **até** *que os centróides não mudem muito ou um limite seja atingido*;
- 7 Atribua cada objeto ao seu centróide mais próximo e retorne os centróides e grupos.

auto-organização emergente muito recente. Argumenta-se que seja especialmente útil para visualizar conjuntos de dados esparsos e de alta dimensão, produzindo uma visão intuitiva de sua estrutura. Uma SOM emergente difere de uma SOM tradicional em que um número muito grande de neurônios (pelo menos alguns milhares) é usado (ULTSCH; HERMANN, 2005). Diz-se que a preservação de topologia da projeção tradicional da SOM é de pouca utilidade quando se usa mapas pequenos: o desempenho de uma pequena SOM é quase idêntica ao de um K-means, com  $k$  igual ao número de nós no mapa (ULTSCH; MOERCHEN, 2005). Uma vantagem adicional de uma ESOM é que ela pode ser treinada diretamente no conjunto de dados disponível sem primeiro ter que executar um procedimento de seleção de recursos (ULTSCH, 2003). Os mapas ESOM podem ser criados e utilizados para análise de dados por meio da ferramenta *Databionics ESOM Public*<sup>9</sup>. Esta ferramenta permite ao utilizador construir mapas ESOM planos e não vinculados (isto é, toroidais) (ULTSCH, 2007).

A Figura 23 é o resultado da aplicação da técnica ESOM, onde é possível visualizar a similaridade de músicas. De uma coleção de músicas é extraído um mapa topográfico a partir de uma característica da música (por exemplo, gênero), onde os pontos de mesma cor são similares.

<sup>9</sup><http://databionic-esom.sourceforge.net/>

Figura 23 – Mapa topográfico com pequenos pontos para as músicas



Fonte: (MARBURG, 2005)

### 3 SOLUÇÕES EXISTENTES

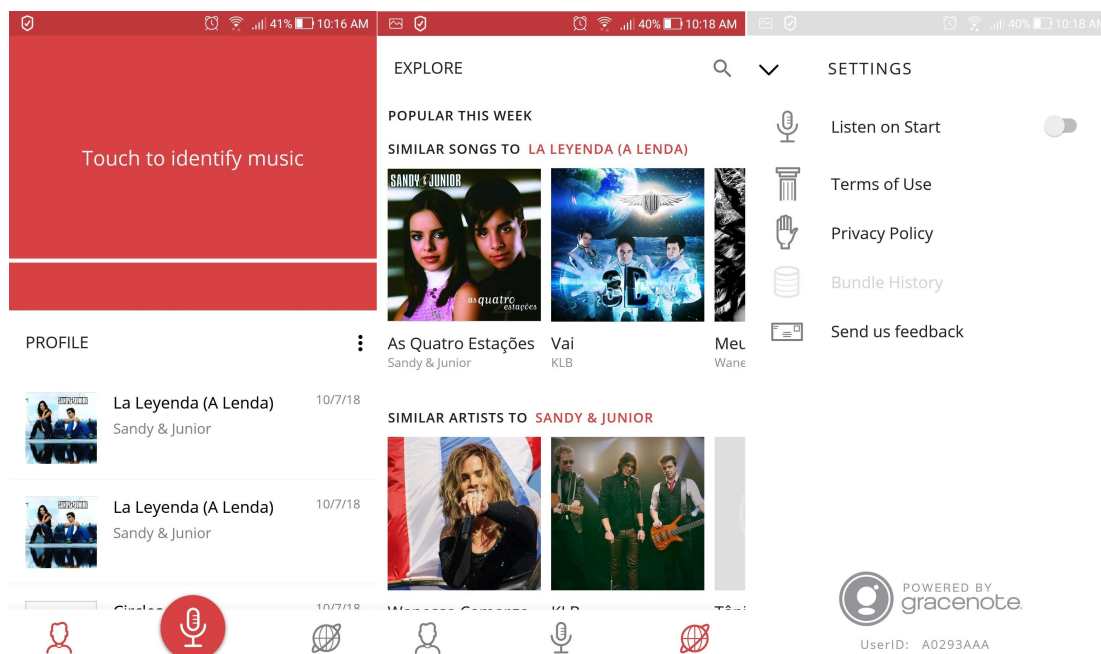
Nos últimos anos, várias plataformas digitais de *streaming* têm surgido, derivado das intensas procuras por música on-line pelos usuários. Este capítulo apresenta de forma resumida as principais soluções comerciais e da academia, para busca de dados musicais.

#### 3.1 SOLUÇÕES COMERCIAIS

##### 3.1.1 MusicID

*Gracenote Inc.*, fundada em 1998, é uma empresa que fornece metadados de música, vídeo, esportes e tecnologias de reconhecimento automático de conteúdo para empresas e serviços de entretenimento em todo o mundo. A solução da empresa destinada à busca de dados musicais se chama *Gracenote MusicID®* disponível para *smartphones* (ver Figura 24).

Figura 24 – MusicID



Fonte: Elaborado pela autora

Segundo o site da companhia<sup>1</sup>, tradução nossa:

O Gracenote MusicID® é o padrão para reconhecimento de música. Ele ajuda os fãs a desbloquear seus álbuns e faixas favoritos na nuvem e a

<sup>1</sup><http://www.gracenote.com/music/music-recognition/>

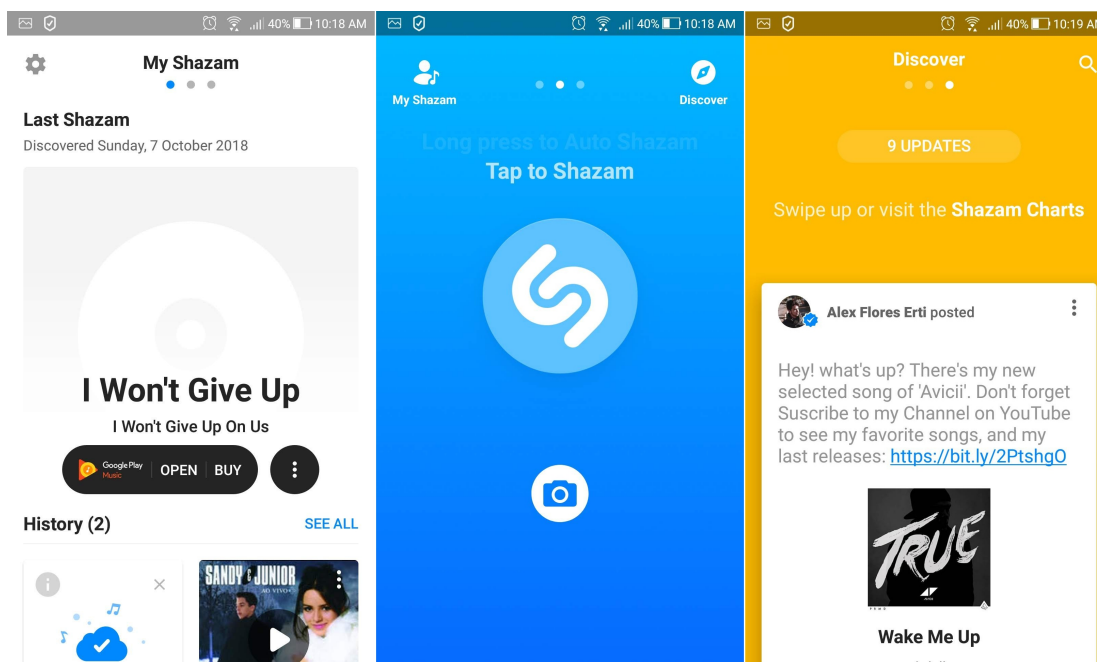
descobrir novas músicas com seus celulares, além de permitir o monitoramento de músicas para detentores de direitos e profissionais do setor (GRACENOTE, 1998).

O Gracenote MusicID®, faz o reconhecimento de músicas que são tocadas ao seu redor, combinado ao uso de *fingerprints* (ver subseção 2.5.1.1) e correspondência de texto para identificar arquivos de música digital em um banco de dados mundial de informações musicais. Uma vez reconhecidos, os arquivos são organizados por nome de faixa, nome do álbum e caminhos de pastas e, então, apresentados ao usuário. Ele é um aplicativo para *smartphones* que identifica músicas ouvindo você cantar, cantarolar ou de músicas que são tocadas ao seu redor.

### 3.1.2 Shazam

*Shazam Entertainment Ltd.* foi fundada em 2000 com a idéia de prover um serviço que pudesse conectar as pessoas à musica, permitindo a identificação da música através de *smartphones*. A aplicação (ver Figura 25) usa o microfone do *smartphone* ou do computador para capturar uma pequena amostra de música e, então, realiza a identificação da música em um grande banco de dados com mais de 12 bilhões de músicas, com uma alta taxa de acertos.

Figura 25 – Shazam



Fonte: Elaborado pela autora

Segundo o site da companhia<sup>2</sup>:

<sup>2</sup><https://www.shazam.com/pt/company>

Shazam é uma aplicação móvel que reconhece música e conteúdos de TV à sua volta. É a melhor maneira de descobrir, explorar e compartilhar a música e os conteúdos de TV que você mais gosta. Levamos 10 anos para alcançar 1 bilhão de tags, 10 meses para chegar a 2 bilhões, 3 meses para ir de 10 a 12 bilhões... É uma aplicação fantástica, agora disponível nas lojas da Apple e Android. E estamos sempre à procura de novas maneiras de encantar os nossos usuários (SHAZAM, 2000).

Para o trecho de música capturado pela aplicação é criado uma *fingerprint* (ver subseção 2.5.1.1), que é comparada com todas as outras *fingerprints* derivadas das músicas no banco de dados. Se houver uma correspondência, são enviadas informações da música para o usuário, como artista, álbum e título da música.

### 3.1.3 SoundHound

*SoundHound Inc.*, fundada em 2005, é uma empresa pioneira em desenvolvimento de aplicações para reconhecimento de voz, compreensão da linguagem natural, reconhecimento de som e tecnologias de busca.

Segundo o site da companhia<sup>3</sup>, tradução nossa:

Acreditamos em permitir que humanos interajam com as coisas ao seu redor da mesma forma como interagimos entre nós: falando naturalmente com telefones celulares, carros, TVs, caixas de música, máquinas de café, e todas as outras partes emergentes do mundo "conectado". Nosso produto mais recente, Hound, utiliza a nossa tecnologia *Speech-to-Meaning*<sup>TM</sup> para mostrar uma experiência inovadora com os *Smartphones*. Nosso produto *SoundHound* aplica nossa tecnologia a música, permitindo as pessoas descobrir, explorar e compartilhar música ao seu redor, e até mesmo encontrar o nome daquela música presa em suas cabeças cantando ou cantarolando. E através da plataforma *Houndify*, capacitamos os desenvolvedores para fazerem parte dessa revolução *speech-to-meaning* (SOUNDHOUND, 2005).

A plataforma independente de Inteligência Artificial *Houndify*, combinada ao *Automatic Speech Recognition* (ASR) e o *Natural Language Understanding* (NLU), permite ao *SoundHound* a identificação de músicas de forma rápida e eficiente. Seus dois produtos conhecidos no meio musical são:

1. *SoundHound Music Search & Play*<sup>4</sup>: aplicativo para *smartphones* onde é possível descobrir, pesquisar e reproduzir qualquer música com controle de voz (ver Figura 26). Ele é um aplicativo para *smartphones* que identifica músicas ouvindo você cantar, cantarolar ou de músicas que são tocadas ao seu redor.

---

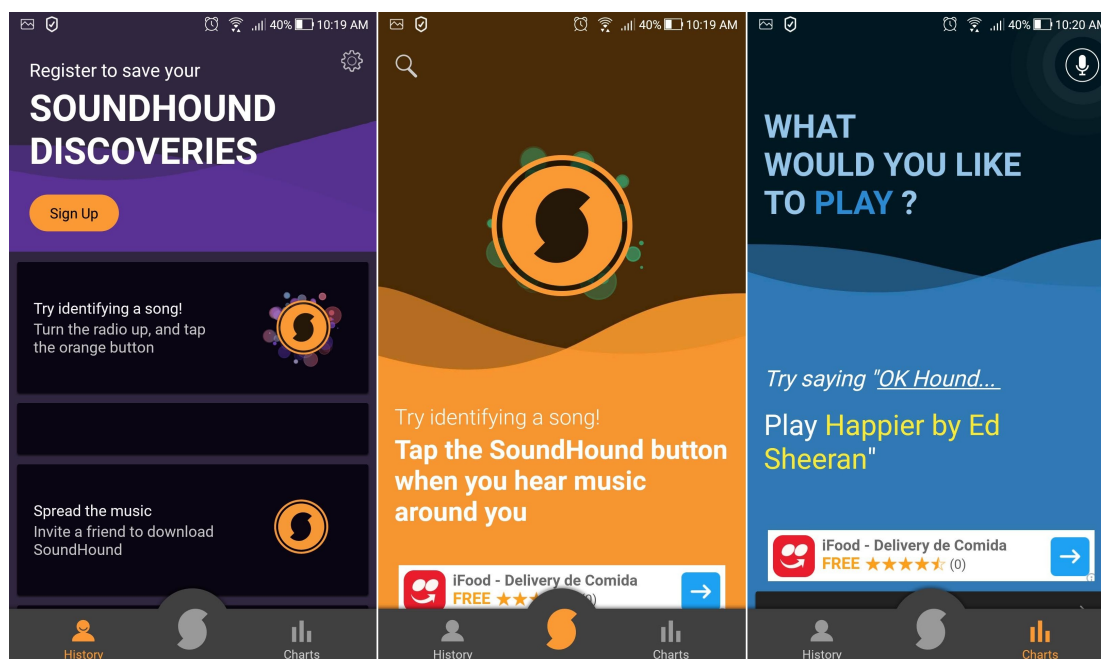
<sup>3</sup><https://soundhound.com/about>

<sup>4</sup><https://soundhound.com/soundhound>



2. *Midomi*<sup>5</sup>: aplicação com as mesmas características do item anterior, porém possui versão para *web*. Sua versão *mobile* é destinada a modelos mais antigos de *smartphones*.

Figura 26 – SoundHound



Fonte: Elaborado pela autora

### 3.1.4 Deezer

Esta solução nasceu da necessidade de facilitar a vida de seu fundador para ouvir e realizar o *download* de músicas. Com isso, o idealizador da plataforma desenvolveu o *Blogmusik.net* em 2006. Devido a sua popularidade, houve objeção de detentores de direitos autores, o que gerou o fechamento do site. Pouco tempo depois, um acordo foi assinado e o antigo site voltou ao ar com o nome de *Deezer* (ver Figura 27).

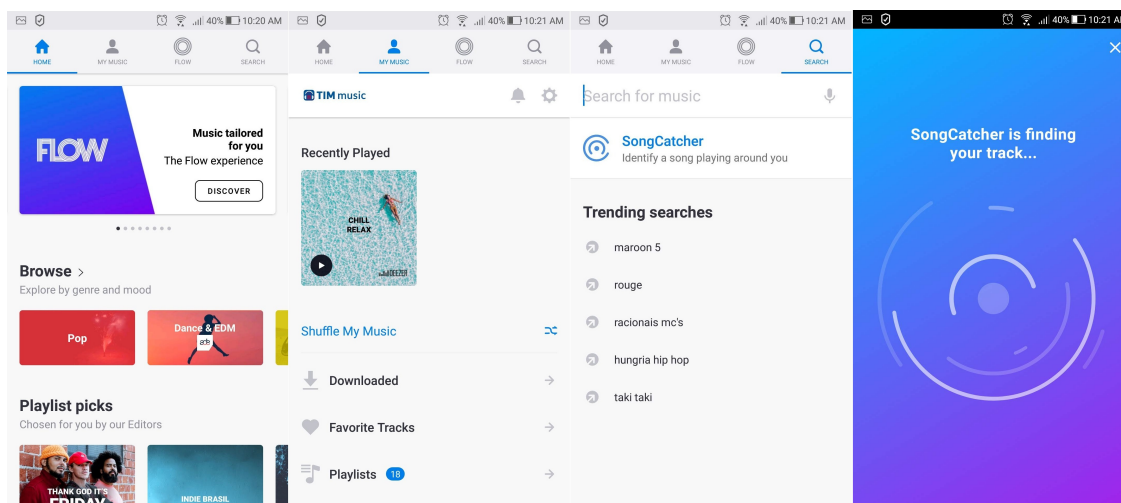
Segundo o site da companhia<sup>6</sup>:

Na Deezer você ouve toda e qualquer música, na hora que quiser. Explore mais de 53 milhões de faixas (e a contagem continua) e descubra artistas e músicas que você vai amar com a recomendação personalizada dos Editores Deezer. A Deezer está em todos os seus dispositivos, tanto online como off-line, sem limites de escuta. Música na ponta de seus dedos para todos os momentos do seu dia: amanhecer, ir ao trabalho, relaxar, viver a vida...é só dar play! (DEEZER, 2006)

<sup>5</sup><https://www.midomi.com/>

<sup>6</sup><https://www.deezer.com/br/company>

Figura 27 – Deezer



Fonte: Elaborado pela autora

A Deezer (ver Figura 27), também conta com uma série de aplicativos que complementam a experiência musical do usuário. O *Stateeztics*, por exemplo, é um *in-app* exclusivo que traça o perfil musical do usuário e mostra suas estatísticas de consumo a partir do seu histórico sonoro. Outra aplicação disponível é o *Edjing*, que oferece mixagem de músicas com diversas ferramentas de efeitos digitais, além de contar com uma interface bastante intuitiva. Já o usuário que está aprendendo a tocar instrumentos musicais pode contar com o *Chordify*, que reconhece o som que está tocando na Deezer e faz a transcrição automática da harmonia em cifras.

Recentemente, no final do ano de 2017, além da correspondência de texto para identificar arquivos de música digital, a Deezer lançou o seu próprio recurso de identificação de músicas que são tocadas ao seu redor combinado ao uso de *fingerprints*, o *SongCatcher*, desenvolvido pela *ACRCloud* (ver subseção 3.1.8).

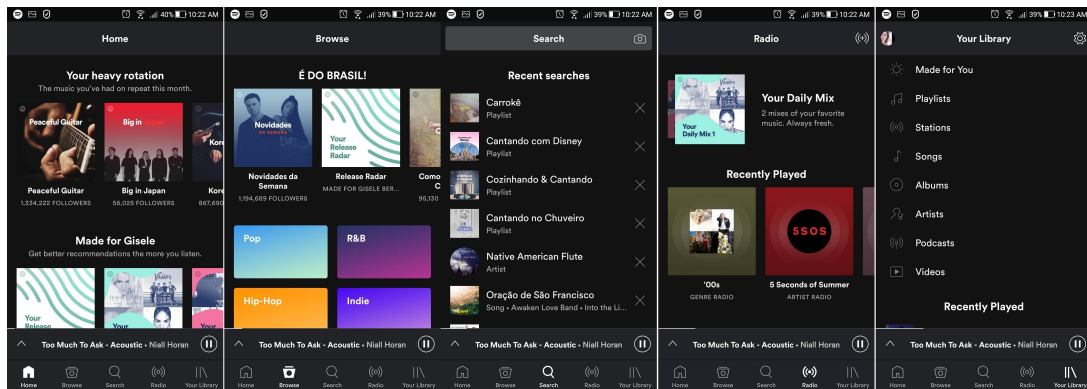
### 3.1.5 Spotify

*Spotify Ltd.*, fundada em 2006, é um serviço de *streaming* de música, *podcast* e vídeo, além de ser o mais usado no mundo. A plataforma fornece conteúdo protegido provido de restrição pela gestão de direitos digitais de gravadoras e empresas de mídia (ver Figura 28). O Spotify é um serviço *freemium*: ele possui recursos gratuitos com propagandas ou limitações, e recursos adicionais, como qualidade de transmissão aprimorada e *downloads* de música, que são oferecidos para assinaturas pagas.

Segundo o site da companhia<sup>7</sup>:

<sup>7</sup><https://www.spotify.com/br/about-us/contact/>

Figura 28 – Spotify



Fonte: Elaborado pela autora

Com o Spotify, é fácil encontrar a música certa para cada momento – no seu telefone, computador, tablet e outros. Existem milhões de faixas no Spotify. Não importa se você está malhando, em uma festa ou relaxando, a música certa está sempre em suas mãos. Escolha o que quer ouvir ou deixe o Spotify surpreendê-lo. Você também pode navegar pelas coleções de músicas de amigos, artistas e celebridades, ou criar uma estação de rádio e simplesmente aproveitar. Produza a trilha sonora de sua vida com o Spotify. Assine ou ouça de graça (SPOTIFY, 2006).

A plataforma emprega um modelo de distribuição de dados híbrido com uma combinação de compartilhamento de dados peer-to-peer<sup>8</sup> (P2P) e uma infraestrutura de servidor. Ao pesquisar uma música através do smartphone e desejar ouvi-lá, o sistema primeiro verifica se a música já se encontra baixada na memória *cache* do *smartphone* para agilizar o processo. Em caso negativo, é feita a conexão diretamente com o servidor do Spotify ao mesmo tempo que o método busca "peers" entre milhões de usuários para que a música que se queira ouvir seja baixada o mais rápido possível. A busca da música é feita pela correspondência de texto para identificar arquivos de música digital através da técnica de recuperação por conteúdo (ver subseção 2.5.1.9).

O Spotify disponibiliza uma *Web API*<sup>9</sup> que permite que desenvolvedores integrem o conteúdo do Spotify em seus próprios aplicativos. O Spotify *Web API* é um serviço com base na arquitetura REST, que retorna em formato JSON dados sobre álbuns, artistas, faixas, playlists, entre outros. Para acessar outras informações é necessária uma autenticação *OAuth*.

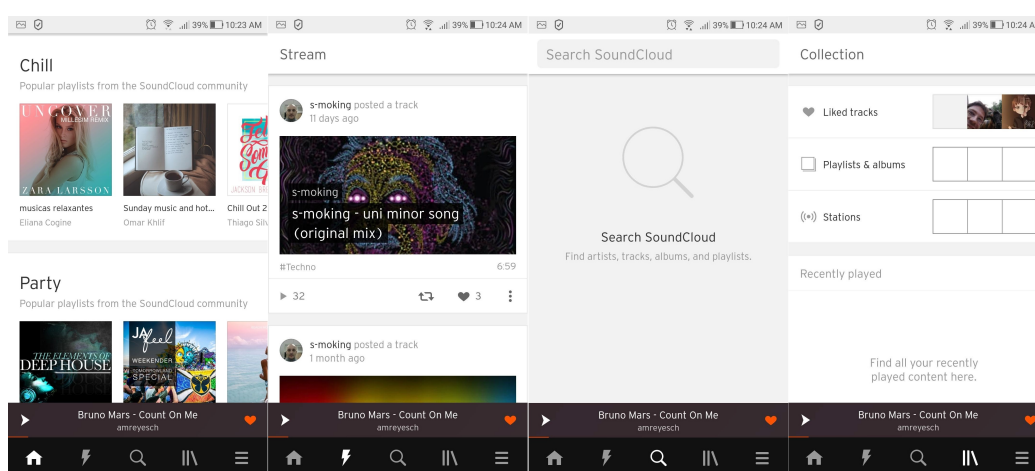
<sup>8</sup>Do inglês par-a-par ou simplesmente ponto-a-ponto, é uma arquitetura de redes de computadores onde cada um dos pontos ou nós da rede funciona tanto como cliente quanto como servidor, permitindo compartilhamentos de serviços e dados sem a necessidade de um servidor central.

<sup>9</sup>Do termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos". É uma forma de integrar sistemas, possibilitando benefícios como a facilidade no intercâmbio entre informações com diferentes linguagens de programação.

### 3.1.6 SoundCloud

*SoundCloud*, criada em 2007, é uma plataforma on-line de publicação de áudio utilizada por profissionais de música (ver Figura 29). Nela os músicos podem colaborar, compartilhar, promover e distribuir suas composições. Originalmente, seu objetivo era permitir que profissionais da música trocassem ideias sobre as composições nas quais estão trabalhando, permitindo uma fácil colaboração e comunicação antes de um lançamento público. Hoje, o site também é utilizado por ouvintes e usuários da *web* em geral.

Figura 29 – SoundCloud



Fonte: Elaborado pela autora

Segundo o site da companhia<sup>10</sup>, tradução nossa:

Como a maior plataforma de música e áudio do mundo, o SoundCloud permite que as pessoas descubram e desfrutem da maior seleção de músicas da mais diversificada comunidade de criadores do mundo. Desde o seu lançamento em 2008, a plataforma tornou-se famosa por seu conteúdo e recursos exclusivos, incluindo a capacidade de compartilhar músicas e se conectar diretamente com artistas, bem como descobrir trilhas inovadoras, demonstrações brutas, *podcasts* e muito mais. Isso é possível graças a uma plataforma aberta que conecta diretamente os criadores e seus fãs em todo o mundo. Os criadores de música e áudio usam o SoundCloud para compartilhar e gerar receita com seu conteúdo com um público global, além de receber estatísticas detalhadas e *feedback* da comunidade do SoundCloud. Ainda não tem uma conta gratuita? (SOUNDCLOUD, 2007)

Os usuários registrados podem ouvir o máximo de conteúdo como quiserem e podem fazer o *upload* de até 180 minutos de áudio ao seu perfil. Todos esses recursos são gratuitos e estão disponíveis para todos os usuários registrados. A plataforma possui uma API

<sup>10</sup><https://soundcloud.com/pages/contact>

integrada a várias aplicações, que permitem fazer o *upload* ou *download* de música e arquivos de música.

O SoundCloud descreve as faixas de música graficamente como formas de onda e permite aos usuários comentar partes específicas do áudio (conhecido como comentários cronometrados). Estes comentários são exibidos ao escutar a parte do áudio que estão se referindo. Outras características incluem respostas, listas de reprodução, seguidores e *downloads* digitais de cortesia. A busca da música é feita pela correspondência de texto para identificar arquivos de música digital através da técnica de recuperação por conteúdo (ver subseção 2.5.1.9). Não foi encontrada documentação informando como funciona detahadamente a recuperação da música.

O SoundCloud também disponibiliza uma *Web API* que permite que desenvolvedores integrem o conteúdo do SoundCloud em seus próprios aplicativos. O SoundCloud *Web API* é um serviço com base na arquitetura HTTP, que retorna em formato JSON dados sobre álbuns, artistas, faixas, playlists, entre outros. Para acessar outras informações é necessária uma autenticação *OAuth*.

### 3.1.7 Musixmatch

A *Musixmatch* foi criada em 2010 com o objetivo de mudar a forma como as pessoas experimentam música e letras. Segundo o site da companhia<sup>11</sup>, tradução nossa:

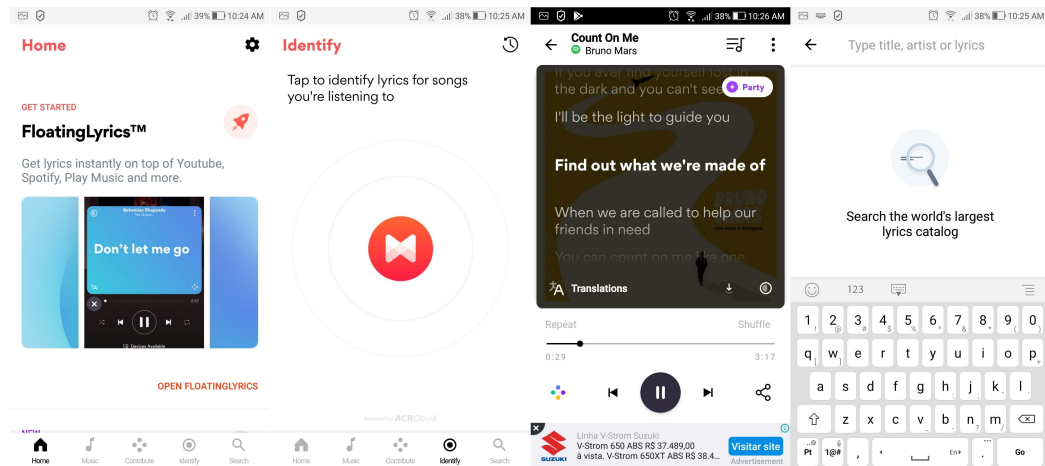
Musixmatch é a maior plataforma de letras do mundo - onde você pode pesquisar, curtir e compartilhar letras de qualquer música, em qualquer lugar do mundo (MUSIXMATCH, 2010).

A plataforma pode ser acessada através do site ou via aplicativo para *smartphones* (ver Figura 30). O Musixmatch digitaliza todas as músicas da biblioteca de música do usuário e encontra letras para todas elas, identificando a letra da música e mantendo sincronizada enquanto a música é tocada. Além da correspondência de texto para identificar arquivos de música digital através da técnica de recuperação por conteúdo (ver subseção 2.5.1.9), ela possui também a capacidade para capturar uma pequena amostra de música através de *fingerprints* (mesma função encontrada em soluções como o *Shazam*), desenvolvido pela ACRCLOUD (ver subseção 3.1.8).

---

<sup>11</sup><http://about.musixmatch.com/>

Figura 30 – Musixmatch



Fonte: Elaborado pela autora

### 3.1.8 ACRCLOUD

A *ACRCloud* foi criada em 2015, sendo a solução vitoriosa no campeonato de *Audio Fingerprinting* do MIREX2015, organizado pelo Laboratório Internacional de Avaliação de Sistemas de Recuperação de Informação Musical (IMIRSEL, sigla em inglês).

ACR (*Automatic Content Recognition*) é uma tecnologia de identificação para reconhecimento de conteúdo reproduzido em um dispositivo de mídia. Ele permite que usuários obtenham rapidamente informações detalhadas sobre o conteúdo que acabaram de experimentar sem qualquer entrada de texto ou esforço de pesquisa.

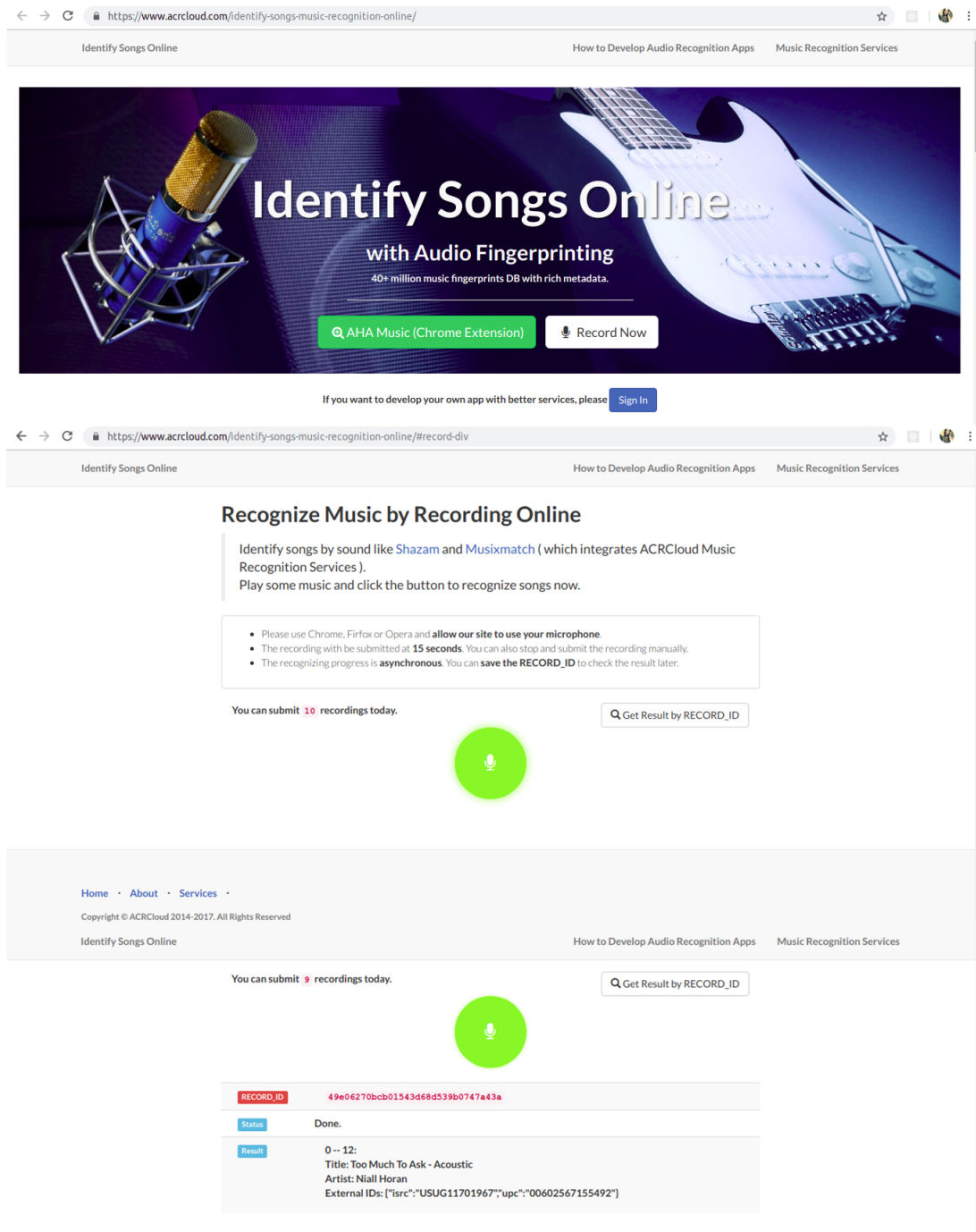
Segundo o site da companhia<sup>12</sup>, tradução nossa:

O ACRCloud fornece serviços ACR em nuvem para ajudar empresas e desenvolvedores excelentes a criar aplicativos baseados em impressões digitais de áudio, como reconhecimento de áudio (suporte a música, vídeo, anúncios on-line e off-line), monitoramento de transmissão, interatividade de segunda tela, detecção de direitos autorais etc (ACRCLOUD, 2015a).

ACRCloud (ver Figura 31) é uma plataforma de micros serviços na nuvem que possui reconhecimento de música através de *fingerprints*, onde identifica músicas ouvindo você cantar ou de músicas que são tocadas ao seu redor, além do monitoramento de transmissão com identificação e apresentação de conteúdo, entre outros. Ele possui integração com serviços de música como o Spotify, Deezer, entre outros, que permite desenvolvedores acessarem diretamente esses serviços e oferecer links diretos para seus usuários.

<sup>12</sup><https://www.acrcloud.com/docs/acrcloud/>

Figura 31 – ACRCLOUD



Fonte: (ACRCLOUD, 2015b), elaborado pela autora

### 3.1.9 Musipedia

Musipedia é uma enciclopédia aberta de música, criação inspirada no Wikipedia<sup>13</sup>, para localização, edição e expansão de coleções de tons, melodias e temas musicais (ver Figura 32). A enciclopédia utiliza o mecanismo de pesquisa de melodias, do qual chamam

<sup>13</sup><https://www.wikipedia.org/>



de *melodyhound*, onde é possível encontrar e identificar uma música, mesmo que a melodia seja tudo o que você saiba no momento. A busca também pode ser feita através da pesquisa de contorno melódico (Código de Pearson) ou com base no ritmo. Ainda, os conteúdos podem ser alterados por qualquer usuário, podendo conter um pedaço de música, um arquivo MIDI, informações textuais sobre o trabalho e o compositor.

A recuperação da informação, segundo o site da enciclopédia<sup>14</sup>, pode ser feita da seguinte forma, tradução nossa:

[...]Você pode tocá-lo em um teclado de piano, assobiar para o computador, simplesmente tocar o ritmo no teclado do computador ou usar o código Pearsons (DEVELOPMENT, 2018?).

É possível também integrar a pesquisa do Musipedia ao seu próprio serviço web, utilizando as interfaces SOAP, que possibilitam pesquisar com base na melodia, no contorno melódico ou no ritmo.

## 3.2 SOLUÇÕES ACADÊMICAS

### 3.2.1 AMUSE

AMUSE (*Advanced Music Explorer*)<sup>15</sup> é um *framework* desenvolvido pela TU Dortmund, na Alemanha, licenciado sob a GPL e implementado em JAVA. Portanto, ele pode ser executado em qualquer sistema operacional que suporte o Java Runtime Environment (ver Figura 33).

Segundo Vatolkin, Theimer e Botteck (2010), o *framework* fornece diferentes funcionalidades, como:

- Processamento de som, convertendo arquivos de áudio MP3 em ondas sonoras;
- *Downsampling* e estéreo para a conversão de arquivos de áudio mono;
- Divisão automática de arquivos wave;
- Escalabilidade usando multi-threading em uma máquina ou fornecendo as tarefas para sistemas de grade como Sun Grid Engine ou LSF Batch;
- Gerenciamento eficiente do conjunto de dados que suporta diretamente o formato WEKA ARFF;
- Componente logger integrado.

<sup>14</sup><https://www.musipedia.org/about.html>

<sup>15</sup><https://sourceforge.net/projects/amuse-framework/>



Figura 32 – Musipedia

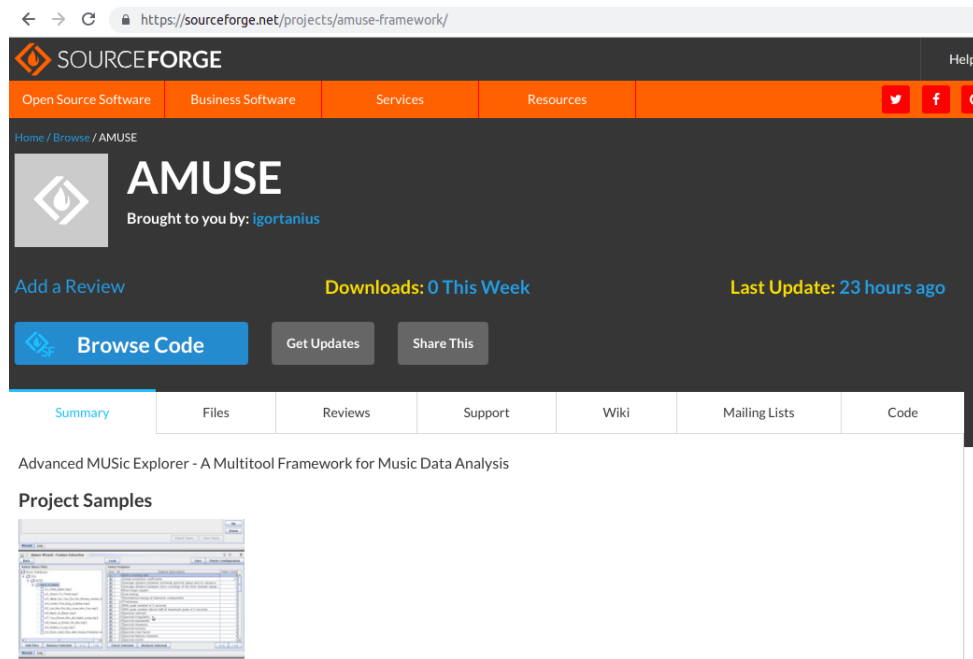


Fonte: (DEVELOPMENT, 2018?), elaborado pela autora

O AMUSE possui subtarefas em cadeia para recuperação da informação musical. Cada tarefa pode ser calculada em várias unidades de processamento. Inicia-se pela tarefa de extração de recursos, que fornece descritores numéricos de baixo nível ou alto nível do sinal de áudio (por exemplo, extração de melodia da música). Depois que a tarefa de extração é carregada na memória, é realizado o processamento dos recursos, em uma etapa intermediária, que serve de entrada para a técnica de classificação (ver subseção 2.5.1.2). Por fim, é realizada a validação dos resultados da classificação.

As ferramentas integradas não têm restrições de uso em relação aos seus códigos-fonte. Se eles não estiverem disponíveis como bibliotecas Java, as versões executáveis

Figura 33 – AMUSE



Fonte: (DORTMUND, 2018?), elaborado pela autora

deverão ser fornecidas. Nesse caso, pode certamente levar à dependência do sistema operacional em execução.

O projeto é oferecido gratuitamente à comunidade de pesquisa. Informações mais detalhadas sobre o projeto podem ser consultadas em (VATOLKIN; THEIMER; BOTTECK, 2010) e (DORTMUND, 2018?).

### 3.2.2 CLAM

CLAM (*C++ Library for Audio and Music*)<sup>16</sup> é um *framework* desenvolvido em C++ no *Music Technology Group* (MTG) da Universidade Pompeu Fabra em Barcelona, Espanha. Ele oferece uma plataforma completa de desenvolvimento e pesquisa para o domínio de áudio e música baseado na técnica SMS (ver subseção 2.5.1.10). Além de oferecer um modelo abstrato para sistemas de áudio, ele também inclui um repositório de algoritmos de processamento e tipos de dados, bem como diversas ferramentas, como entrada/saída de áudio ou MIDI (ver Figura 34).

Segundo Amatriain (2004), as características mais importantes do framework são:

- Verdadeiramente orientado a objetos. Extensas técnicas de engenharia de software foram aplicadas para projetar uma estrutura que seja altamente (re)utilizável e compreensível;

<sup>16</sup><http://clam-project.org/>

Figura 34 – CLAM

The screenshot shows the CLAM project website. At the top, there's a navigation bar with links: Home, Faq, Doc, Wiki, Planet, Download, Screenshots, Contact, Devel, TestFarm, Bugs. Below this, a 'NEWS' section highlights 'May 07, 2013' with the announcement 'TestFarm 2.0 released'. The text describes the release as a major rewrite on the server side, more reliable, scalable, and easier to maintain. It provides instructions on how to install it using pip or as a Debian package. A 'TestFarm' logo is also visible. Below the news, there's a section for 'April 04, 2011' titled 'CLAM at Debian', mentioning that CLAM is now in the official Debian repositories. A 'BUG TRACKER' section is also present, showing a list of issues categorized by status: Unassigned (1-10/14), Resolved (1-10/13), and Recently Modified (1-10/36). The bug tracker interface includes filters for 'Main', 'My View', 'View Issues', 'Change Log', and 'Roadmap'. The 'Unassigned' list shows various bugs related to CLAM modules like 'Tonal Analysis', 'SpectralDescriptorsTest', and 'Processing'. The 'Resolved' list shows bugs that have been fixed, such as 'Unable to build Application: Chordata' and 'Inconsistent navigation on the website'.

Fonte: (DEVELOPMENT, 2004), elaborado pela autora

- É abrangente, uma vez que não só inclui classes para processamento, mas também para entrada e saída de áudio e MIDI, serviços de serialização XML, algoritmos e visualização e interação de dados, e manipulação multi-threading;
- Lida com uma ampla variedade de tipos de dados extensíveis que vão desde sinais de baixo nível (como áudio ou espectro) até estruturas semânticas de nível superior (como frase musical ou segmento);
- É multiplataforma. Todo o código é ANSI C++ e é regularmente compilado no Linux, Windows e Mac OSX usando os compiladores mais usados. Até mesmo o código para entrada/saída, visualização e multithreading é de plataforma cruzada até a camada mais baixa possível;
- O projeto está licenciado sob os termos e condições GPL (Licença Pública GNU). Apesar de possuir a opção de licenciamento duplo da estrutura (ou seja, oferecer uma licença comercial alternativa), tudo o que é oferecido na versão pública é GPL e o projeto é, portanto, Software Livre, código aberto e colaborativo;

- Base para todos os desenvolvimentos futuros no MTG e, portanto, mantido e atualizado regularmente;
- O framework pode ser usado como uma biblioteca C++ regular ou como uma ferramenta de prototipagem. No primeiro modo, o usuário pode estender, adaptar ou otimizar a funcionalidade da estrutura para implementar um aplicativo específico. No segundo modo, o usuário pode facilmente construir um protótipo para testar um novo algoritmo ou aplicativo de processamento de sinais.

Informações mais detalhadas sobre este *framework* podem ser consultadas em (AMATRIAIN, 2007, 2004).

### 3.2.3 Java MIR

jMIR (Java MIR)<sup>17</sup> é um *software* que possui um conjunto de componentes desenvolvido na CIRMMT e Marianopolis College, ambos localizados no Canadá. Cada um dos componentes pode ser utilizado separadamente ou como um todo (ver Figura 35).

O software é de código livre implementado em Java para uso nas pesquisas de Recuperação de Informação Musical (MIR) e se baseia em uma técnica de mineração de dados, a classificação (ver subseção 2.5.1.2). Ele pode ser usado para estudar música na forma de gravações de áudio, codificações simbólicas e transcrições líricas, e também pode extrair informações culturais da Internet. Ainda, ele inclui ferramentas para gerenciar e criar perfis de grandes coleções de músicas e para verificar o áudio quanto a erros de produção. É bem documentado e inclui GUIs para aumentar a usabilidade geral.

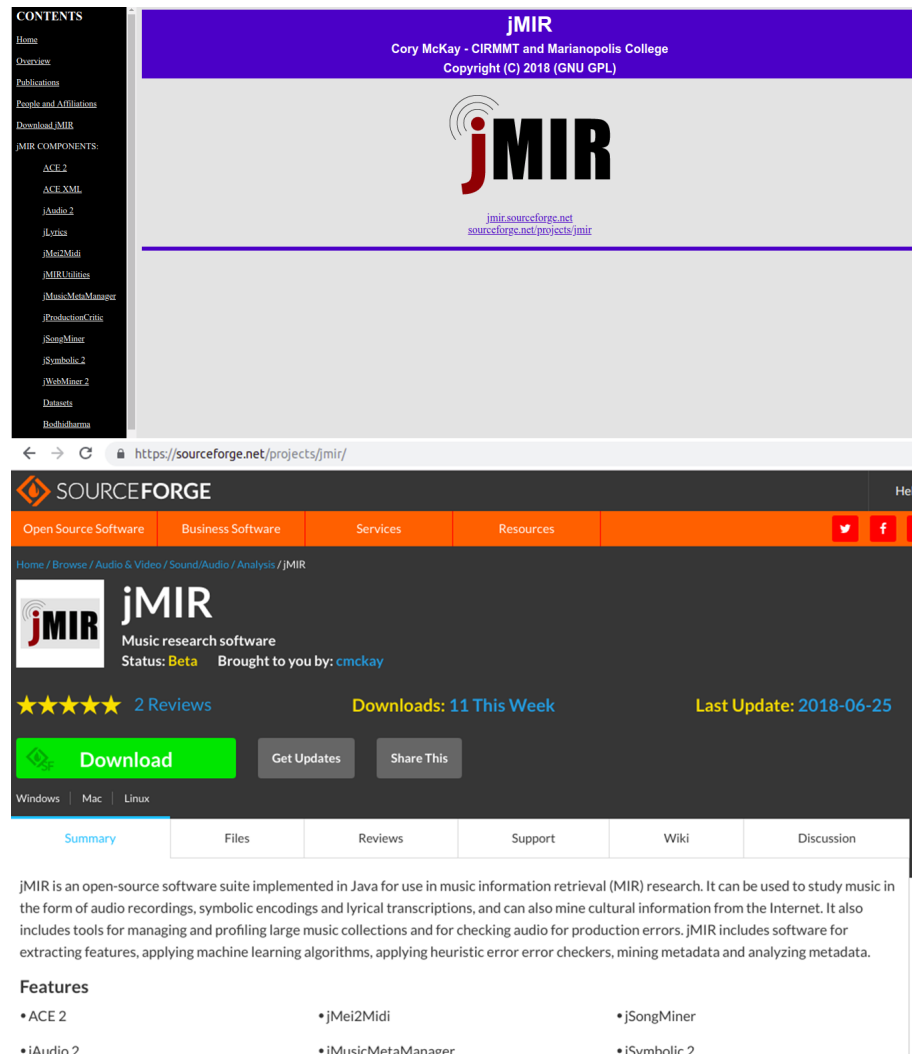
O objetivo principal do *software* é auxiliar nas pesquisas em classificação automática de música e a análise de similaridade, proporcionando as seguintes características:

- Tornar tecnologias sofisticadas de reconhecimento de padrões acessíveis a pesquisadores de música com históricos técnicos e não técnicos;
- Eliminar duplicação redundante de esforço;
- Aumentar a cooperação e a comunicação entre os grupos de pesquisa;
  - Facilitar o desenvolvimento iterativo e o compartilhamento de novas tecnologias MIR;
  - Facilitar comparações objetivas de algoritmos.

---

<sup>17</sup><http://jmir.sourceforge.net/>

Figura 35 – Java MIR



Fonte: (MCKAY, 2018?), elaborado pela autora

- Facilitar a pesquisa combinando características musicais de alto nível, baixo nível e culturais (ou seja, características simbólicas, áudio e web-minadas).

Informações mais detalhadas sobre o projeto estão disponíveis nas publicações acadêmicas<sup>18</sup>. Manuais e documentação para cada componente também podem ser consultados em (MCKAY, 2018?) e (MCKAY, 2010).

### 3.2.4 MIRtoolbox

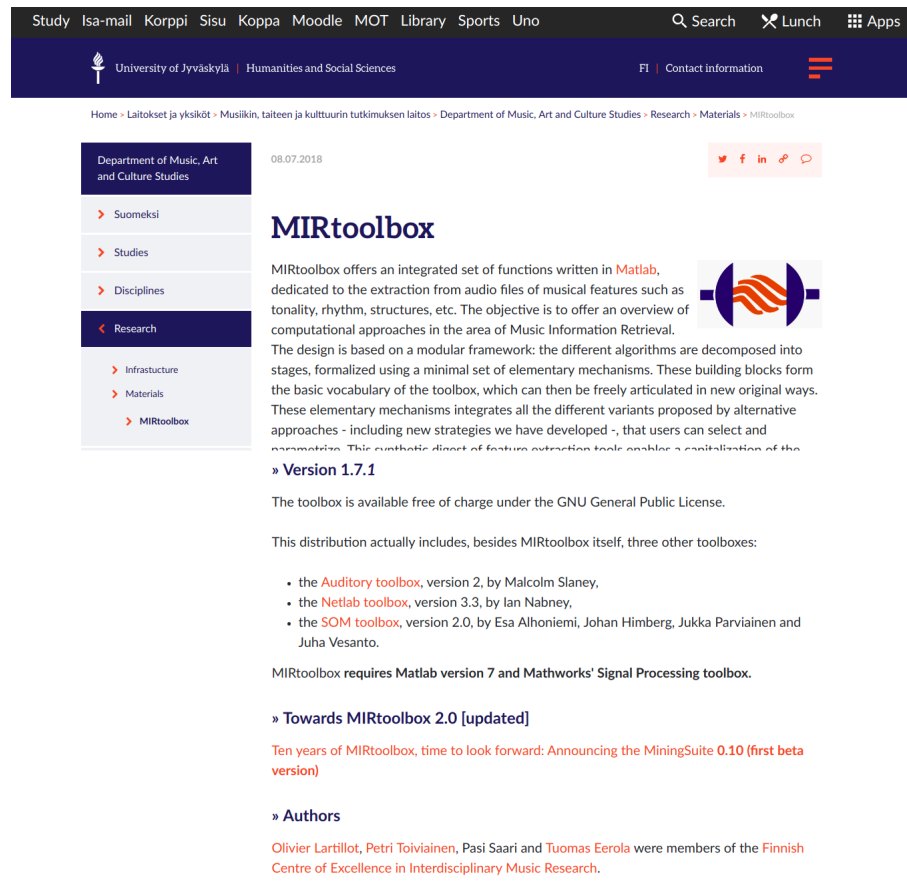
*MIRtoolbox*<sup>19</sup> é um pacote de ferramentas escritas em Matlab para a extração de recursos musicais, como tonalidade e ritmo, tanto para especialistas quanto para não es-

<sup>18</sup><http://jmir.sourceforge.net/publications.html>

<sup>19</sup><https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox>

pecialistas do Matlab (ver Figura 36). Ele foi desenvolvido dentro do contexto de um Projeto Europeu chamado “Tuning the Brain for Music”, dedicado ao estudo da música e da emoção, com colaboração entre neurociências, psicologia cognitiva e ciência da computação. Os grupos e instituições envolvidos são a Music Cognition Team da University of Jyväskylä na Finlândia e o Music Acoustics Group do KTH em Estocolmo.

Figura 36 – MIRtoolbox



Fonte: (JYVÄSKYLÄ, 2018?), elaborado pela autora

Segundo Lartillot (2013), foi elaborado um manual onde são descritas as seguintes especificações da solução:

- Quadro modular: É baseado em um conjunto de blocos de construção que podem ser parametrizados, reutilizados, reordenados, etc.;
- Sintaxe simples e adaptativa: Os usuários podem se concentrar no design geral e o MIRtoolbox que cuida das tarefas laboriosas subjacentes;
- Software livre e código-fonte aberto: A ideia é propor a capitalização da expertise da comunidade de pesquisa e oferecê-la de volta à comunidade;

- Recursos: O MIRtoolbox inclui cerca de 50 extratores de recursos de áudio e música e descritores estatísticos.

O MIRtoolbox é baseado em técnicas de mineração de dados e, por ser um pacote de ferramentas, são variadas as técnicas para recuperação da informação musical, como a classificação (ver subseção 2.5.1.2).

Desta forma, ele pode ser útil para a comunidade de pesquisa em Recuperação da Informação Musical (MIR), mas também para fins educacionais. Mais informações sobre o projeto podem ser consultadas em (LARTILLOT; TOIVIAINEN, 2007; LARTILLOT, 2013) e (JYVÄSKYLÄ, 2018?).

### 3.2.5 MusicMiner

O *Databionic MusicMiner*, desenvolvido como parte de um projeto de pesquisa do Grupo de Pesquisa em Databionics da Universidade de Marburg, na Alemanha, é um navegador para dados musicais baseado em técnicas de mineração de dados, como clusterização (ver subseção 2.5.1.3) e visualização (ver subseção 2.5.1.11) com base no paradigma dos mapas geográficos ESOM (ver Figura 37).

A coleção de músicas é recuperada e apresentada em forma de mapa topográfico com pequenos pontos para as músicas. O usuário pode interagir com o mapa de diferentes formas para selecionar e ouvir músicas; explorar suas músicas; e criar playlists baseadas no paradigma de mapas geográficos.

O site do projeto<sup>20</sup> apresenta as seguintes características da solução:

- Análise automática de uma árvore de pastas com arquivos de música (MP3, OGG, WMA, M4A, MP2, WAV);
- Descrição automática de arquivos de áudio digital por som;
- Criação de *MusicMaps* para navegar pelo espaço sonoro com base no paradigma dos mapas geográficos ESOM;
- Criação visual de *playlists*;
- Pesquisa por similaridade na coleção de músicas com base no som;
- Navegação hierárquica personalizável da base de dados, como por exemplo, por gênero/artista/álbum ou ano/artista;

---

<sup>20</sup><http://musicminer.sourceforge.net/>

Figura 37 – MusicMiner

The screenshot shows the MusicMiner website. The header includes the MusicMiner logo and the text 'Databionics Research Group'. The main content area is divided into several sections: 'News' with a list of updates, 'Databionics MusicMiner' with a description of the tool, and 'Usage of MusicMiner audio features' with detailed instructions on how to use the tool and a table of feature sets.

**News**

- 04/06/2006: New features to generate a harmonic playlist and to synchronize the playlist with a mp3 player in 0.9.6. Also many fixes and additions.
- 02/22/2006: New function to share audio features and rate songs via a webservice in 0.9.5. Also many fixes and additions.
- 11/23/2005: Minor release 0.9.1 fixing music playing under Windows.
- 11/09/2005: Mailing lists are finally fixed.
- 08/15/2005: Added partly untested tutorial for compiling MusicMiner on Max OS X contributed by Christopher Suckling.
- 07/29/2005: MusicMiner 0.8.4: Better audio processing with MPlayer, including support for \*.wma and \*.m4a formats.
- 07/22/2005: Please report bugs in the tracker and ask for help in the forums until the mailing lists are fixed.
- 07/20/2005: MusicMiner 0.8.3: Fixes database startup problem reported on some systems.
- 07/14/2005: MusicMiner 0.8.2: Fixes problem with spaces in install path.
- 07/13/2005: MusicMiner 0.8.1: Fixes several issues.
- 07/09/2005: MusicMiner 0.8.0 released!

**Databionics MusicMiner**

The Databionics MusicMiner is a browser for music based on data mining techniques. You can create MusicMaps to visualize the similarity of songs and artists. Explore your music and create playlists based on the paradigm of geographical maps! Features include:

**Usage of MusicMiner audio features**

The prototypes of the audio feature used in the publications were written in Matlab. Due to limited programming resources it was impossible to create an exact reproduction in Java. Instead we repeated the methodology described in the publications with Java using a slightly different set of low level features and a much larger set of high level features. The resulting top 20 features are defined in etc/features/musicminer-VERSION.xml and they showed similar performance to the original 20 features defined e.g. in our technical report.

If you want to extract these features on your own music datasets you need to

1. Install MusicMiner
2. Use e.g. `mmadd -f ~/music/mp3 -r` to add all music files in the specified folder and all subdirs to the database. Use the option `-g` to force a genre, otherwise it is read from the ID3 tag.
3. Use e.g. `mmafe -f musicminer -s 30 -x ~/music/feat` to extract the MusicMiner audio features from a 30s segment in the center of each song.

The results are stored in two files per song in the specified folder named with a unique key per song. The \*.ln files contain the unique key and the 20 features in a single tab separated line. The \*.names files contain the original filename and the genre (if available) separated with tabs. The file `musicminer.names` contains the names of the features. You can add more songs later with `mmadd` and rerun `mmafe`. The extraction works incrementally, the program checks for existing files and only processes songs where no features have been extracted, yet. Note, that when using the `-x` option the features are only stored in the files and not in the database.

For further processing you can e.g. use the Unix commands `cat [0-9]*.ln > songs.ln` and `cat [0-9]*.ln > songs.names` to merge the single files and import them into your machine learning tools (for Windows see GNU utilities for Win32). Note, that the raw feature values are stored. For clustering and distance calculations a normalization e.g. to zero mean and unit variance should be applied.

The option `-k` can be used with `mmafe` to save the downsampled (mono, 22050Hz), trimmed, and normalized (DC 0, maximum absolute amplitude = 1) \*.wav files in the same location as the original sound files with changed file extensions. This is useful to ensure the same conditions when extracting audio features with different programs for comparison.

Other feature sets can be defined using the Yale XML syntax. For some examples see etc/features/\*.xml:

Name	Description	Size
musicminer	Default MusicMiner audio features	20
generate40k	Large set of candidate audio features	39,760
generate688k	Huge set of candidate audio features	687,960
mfcc	Mean and standard deviation of first 20 MFCC	40
chroma	Mean and standard deviation of 12 Chroma tones	24

Fonte: (MARBURG, 2005), elaborado pela autora

- Base de dados flexível, incluindo o armazenamento separado de vários artistas por música, álbuns e listas de reprodução como parte de uma lista de reprodução;
- Importação e exportação de meta informações baseadas em XML.

O *MusicMiner* é escrito em Java para máxima portabilidade e publicado sob os termos da GPL (General Public License). Seu foco principal é a pesquisa e o ensino. Informações mais detalhadas sobre o projeto podem ser encontradas em (MORCHEN et al., 2005) e (MARBURG, 2005).

### 3.2.6 Tunebot

*Tunebot*<sup>21</sup> é um projeto criado em 2015, desenvolvido e mantido por *Interactive Audio Lab*<sup>22</sup> na Universidade de Northwestern, nos Estados Unidos. Segundo Huq, Cartwright e Pardo (2010b), o Tunebot está disponível como um serviço web (ver Figura 39) e

<sup>21</sup><http://music.cs.northwestern.edu/data/tunebot/>

<sup>22</sup><http://music.eecs.northwestern.edu/>



está atualmente em teste beta como um aplicativo do iPhone (ver Figura 38). A interação do usuário nas versões da Web e do iPhone é idêntica: (i) cante, e (ii) escolha. O usuário simplesmente canta uma parte da música desejada para o Tunebot e o sistema retorna uma lista ordenada de músicas. Cada música pode ser reproduzida por um simples clique. Enquanto a música está tocando, o sistema apresenta uma caixa de diálogo perguntando se esta é a música correta. Se o usuário clicar em "sim", a consulta será armazenada no banco de dados como um exemplo pesquisável para essa música. O usuário é conectado à Amazon.com ou ao iTunes, onde a música pode ser comprada.

Figura 38 – Aplicativo Tunebot



Fonte: (LAB, 2008a), elaborado pela autora

O sistema não exige chaves de pesquisa codificadas manualmente, pois atualiza automaticamente o banco de dados com novas chaves de pesquisa derivadas de consultas e contribuições do usuário (HUQ; CARTWRIGHT; PARDO, 2010b). O banco de dados do *Tunebot* compara as músicas com as músicas cantadas pelos usuários utilizando o algoritmo *query by humming* (ver subseção 2.5.1.8).

Outro objetivo do projeto é ajudar pesquisadores na área de reconhecimento de músicas que utilizam o algoritmo *query by humming*, facilitando uma pesquisa mais precisa do desempenho do mundo real do que seria possível com conjuntos de dados existentes.

Informações mais detalhadas sobre o projeto podem ser consultadas em (HUQ; CARTWRIGHT; PARDO, 2010b; CARTWRIGHT; PARDO, 2012) e (HUQ; CARTWRIGHT; PARDO, 2010a).

Figura 39 – Tunebot Web



Fonte: (HUQ; CARTWRIGHT; PARDO, 2010b)

## 4 ANÁLISE COMPARATIVA

Esta seção apresenta os critérios definidos para a análise das soluções existentes, a análise comparativa propriamente dita e os resultados obtidos com a análise. A metodologia de comparação entre as as soluções leva em conta documentações públicas disponíveis que possuam informação pertinente.

### 4.1 CRITÉRIOS DE ANÁLISE

Com base nos conceitos apresentados por Wazlawick (2012, p.305), que define os atributos de qualidade internos, externos e de uso de produtos de software; e pela autora Wangenheim (2017), que apresenta heurísticas definidas por Nielsen (1995) para assegurar que os produtos são usáveis, esta seção tem por objetivo apresentar os critérios que devem ser considerados na avaliação das soluções para busca de dados musicais, de modo a compará-las e facilitar a escolha pela mais apropriada para uma determinada situação.

Neste trabalho foi realizada uma adaptação de heurísticas para permitir uma boa análise da eficiência e adequação funcional das soluções apresentadas, sendo utilizado informações disponibilizadas nas documentações próprias de cada solução comercial e acadêmica. Já para a análise da usabilidade foi realizada uma adaptação do *MATch Checklist* disponibilizado pelo Grupo de Qualidade de Software (GQS) da UFSC, onde o conjunto de perguntas possui uma escala de resposta com 2 opções: Sim (a solução atende o objetivo) e Não (a solução não atende o objetivo).

Assim sendo, os seguintes critérios e subcritérios são considerados:

- **Eficiência:** Trata da otimização do uso de recursos de tempo e espaço. Espera-se que o sistema seja o mais eficiente possível de acordo com o tipo de problema que ele soluciona:
  - *Comportamento em relação ao tempo:* Mede o tempo que o sistema leva para processar suas funções, ou seja, o tempo de reconhecimento/busca de uma música;
  - *Utilização de recursos:* Avalia a complexidade das estratégias e algoritmos utilizados na recuperação de informação musical;
  - *Bitrate:* Mede a qualidade do áudio. Essa qualidade consiste no número médio de bits que será comprimido em um segundo de dados. A unidade utilizada é o KBPS ou 1000 BITS por segundo;

- **Adequação Funcional:** Mede o grau no qual o produto oferece funções que satisfazem necessidades estabelecidas e implicadas quando o produto é usado sob condições especificadas:
  - *Disponibilidade:* Avalia a disponibilidade da aplicação em diferentes plataformas;
  - *Modelo de desenvolvimento:* Avalia se a solução é de código aberto, dando a possibilidade para que qualquer um consulte, examine ou modifique o produto;
  - *Integrações:* Avalia se a solução permite extensões e/ou integrações com outras aplicações;
  - *Acessibilidade:* Avalia se a solução possui acesso ao acervo de músicas on-line e/ou off-line;
  - *Busca de dados:* Avalia se a solução foi projetada para *matching* exato ou por similaridade (aproximado);
  - *Inclusão da dados:* Avalia se a solução permite o envio de músicas feito pelo usuário;
  - *Modelo de pagamento:* Avalia o custo da solução, como por exemplo: Gratuito, Pago ou Freemium.
- **Usabilidade:** Avalia o grau no qual o produto tem atributos que permitem que seja entendido e que seja atraente ao usuário, quando usado sob condições especificadas:
  - *Visibilidade do status do sistema:* O sistema deve sempre manter o usuário informado sobre o que está acontecendo. Por exemplo, os componentes interativos selecionados são claramente distintos dos demais?
  - *Prevenção de erros:* Mensagens de erros devem ser claras e objetivas, devem indicar o problema com precisão e sugerir uma solução;
  - *Flexibilidade e eficiência de uso:* A solução deve permitir configuração de ações frequentes. Por exemplo, as funções mais utilizadas são facilmente acessadas?
  - *Estética e Design minimalista:* Mensagens de diálogos não devem conter informações irrelevantes. Por exemplo, o menu é esteticamente simples e claro, com opções fáceis de encontrar, dispostas em uma ordem lógica e com títulos curtos?
  - *Pouca interação homem/dispositivo:* Qualquer informação deve ser fácil de pesquisar e deve ser focada na tarefa do usuário. Por exemplo, a navegação da solução é intuitiva, é fácil chegar à tela desejada?

## 4.2 ANÁLISE

Uma comparação é primeiramente realizada entre as soluções comerciais, seguida das soluções acadêmicas.

### 4.2.1 Soluções Comerciais

Inicialmente, para a análise das soluções comerciais, foram criadas 3 tabelas, uma para cada grande critério analisado para melhor visualização.

Em relação ao critério de Eficiência mostrado na Tabela 4, o comportamento das soluções em relação ao tempo foram constatados as diferenças verificando o tipo de método utilizado para o reconhecimento/busca de músicas. Os testes foram realizados usando conexão de Internet Wi-Fi e Dados Móveis (4G). Com um smartphone e um cronômetro, foram feitas 5 execuções em cada aplicação com músicas diferentes. E então, calculado a média de todos os tempos para cada aplicação.

Para as soluções utilizando RPC, foi testada a busca de músicas pela correspondência de texto através de metadados (ou seja, através da informação do título, álbum ou gênero da música). Neste caso, a busca demorou até 30s para retornar uma amostra de resultados. Para as soluções que utilizam *Fingerprint* e fazem reconhecimento de música de forma exata (E), foi testada a busca através da informação de uma parte de áudio "original". Já para as soluções de reconhecimento de músicas de forma aproximada (A), foi testada a busca através da informação de uma parte de áudio "original", do canto ou do cantarolar. Ambas as soluções podem variar de 5s até 13s para retornar uma amostra de resultados.

Cabe observar aqui que uma análise feita apenas sobre o comportamento geral das soluções em relação ao tempo não seria relevante, já que tempos entre 5s e 30s são relativos à situação em que o usuário está no momento. Por exemplo, o usuário pode estar em lugares onde a conexão de Internet não é boa ou quase nula, ou pode estar em lugares onde a conexão de Internet é ótima, já que o reconhecimento de uma música depende de uma conexão de Internet. Quanto melhor a conexão, mais rápido será o reconhecimento da música.

É possível verificar que tanto soluções para o reconhecimento de músicas de forma aproximada, como o MusicID e o ACRCLOUD, quanto soluções para o reconhecimento de forma exata, como o Shazam, Deezer e Musixmatch, podem utilizar o mesmo método de *Fingerprint*. Apesar de utilizarem o mesmo método, a forma como pode ter sido desenvolvido é que gerou a vantagem competitiva dentre os concorrentes do mesmo ramo.

O SoundHound utiliza a tecnologia de Inteligência Artificial (IA) para o reconhe-

cimento de músicas. O desenvolvimento de uma solução baseada em IA se dá através de métodos e algoritmos de aprendizagem de máquina, como Classificação e Clustering, porém não foi encontrada documentação que especificasse o método ou algoritmo utilizado pelo SoundHound para o reconhecimento das músicas. Com essa solução, é possível recuperar músicas através da voz, cantarolando ou até mesmo informando uma parte do áudio "original". Conforme os usuários cantam e conforme a escolha da música do resultado amostrado, a IA associa a cantoria àquela música. Então, quanto mais cantar e buscar as músicas "certas" para a cantoria, a taxa de acerto aumenta, formando uma rede totalmente interligada para o reconhecimento correto de músicas.

Há também as soluções que utilizam o método de Recuperação por Conteúdo (RPC), que seriam as cadeias de caracteres, em buscas, por exemplo, por título, álbum ou gênero da música - o que chamamos de metadados -, conforme já explicitado em seções anteriores deste trabalho. Destas soluções, pode-se citar o Spotify e o SoundCloud. A busca das músicas no Spotify, por exemplo, é feita exclusivamente através de texto, não sendo possível adicionar músicas criadas pelo usuário. Entretanto, é possível criar playlists com as suas músicas preferidas.

Para o Musipedia, não foi encontrada documentação que especificasse o método ou algoritmo utilizado para o reconhecimento das músicas.

Quanto ao critério de Bitrate, das 9 soluções analisadas, foi encontrado essa informação em apenas 4. Diferentes bitrates podem produzir qualidade de som variável. Assim, quanto maior o bitrate, mais vezes por segundo o som original é amostrado, produzindo assim uma reprodução mais autêntica e melhor do som (SANDISK, 2018?).

O Deezer e Spotify apresentam bitrate de até 320kbps, possuindo 3MB por minuto de áudio e uma qualidade aproximada à de um CD. Já o MusicID e o SoundCloud apresentam bitrate de até 128kbps, possuindo 1MB por minuto de áudio e uma qualidade mínima aceitável em termos de alta fidelidade do áudio (F, 2012).

Em relação ao critério de Adequação Funcional mostrado na Tabela 5, das 9 soluções analisadas, 7 são multiplataformas, sendo em sua maioria voltadas para uso em smartphones e 2 de uso exclusivo via Web.

Pode-se verificar que as soluções comerciais, em sua grande maioria, possuem um modelo de desenvolvimento fechado (F), ou seja, seu código não pode ser alterado, mas disponibilizam APIs Web para a comunidade de desenvolvedores, para que a solução possa ser incorporada a seus próprios sites e aplicações.

Quanto à acessibilidade, todas precisam de conexão com a Internet para o uso de suas funcionalidades e apenas 2 das 9 soluções analisadas possuem acesso off-line, que se

Tabela 4 – Análise das soluções comerciais: Critério de Eficiência

CRITÉRIOS		Eficiência		
		Comportamento em relação ao tempo	Utilização de recursos	Bitrate
C	<b>MusicID</b>	Até 8s	FP	Até 128kbps
O	<b>Shazam</b>	Até 8s	FP	-
M	<b>SoundHound</b>	Até 5s	IA	-
E	<b>Deezer</b>	Até 10s	RPC e FP	Até 320kbps
R	<b>Spotify</b>	Até 30s	RPC	Até 320kbps
C	<b>SoundCloud</b>	Até 30s	RPC	Até 128kbps
I	<b>Musixmatch</b>	Até 13s	RPC e FP	-
A	<b>ACRCloud</b>	Até 5s	FP	-
L	<b>Musipedia</b>	-	-	-

Legenda: FP - Fingerprint; IA - Inteligência Artificial; RPC - Recuperação por Conteúdo;

dá através do download de músicas para a memória interna do dispositivo. As 2 soluções são Spotify e Deezer.

Das 9 soluções comerciais analisadas, 4 possuem reconhecimento de músicas de forma aproximada (A), ou seja, é possível realizar buscas de músicas por similaridade através da voz, sendo que 2 delas permite cantarolar. Já as 5 demais soluções realizam o reconhecimento de músicas de forma exata (E), sendo necessária uma parte de áudio "original" e/ou uma busca por conteúdo através de metadados.

Apenas 3 soluções permitem a inclusão de músicas criadas por usuários, enquanto as outras 6 permitem a inclusão de novas músicas somente através de contatos com gravadoras e/ou artistas. Destas 3 primeiras, duas reconhecem músicas de forma aproximada: ACRCloud e Musipedia. A primeira é um serviço na nuvem, sendo, até o momento, o maior banco de dados de músicas e um serviço utilizado pela maioria das outras soluções aqui analisadas, como Musixmatch, que permite o reconhecimento de músicas por similaridade, e o Musipedia é uma wikipedia de músicas que aceita contribuições musicais de diversas formas: através da voz, partes de músicas, ou até em formato MIDI (quando o som é criado digitalmente).

Com exceção do ACRCloud, todas as demais soluções possuem versões gratuitas

para uso, com a possibilidade de pagar uma mensalidade e não ter interrupções e propagandas entre as músicas. Ainda, poucas soluções possuem integrações com outros serviços e/ou aplicações. Com o Shazam, por exemplo, é possível integrar-se ao Spotify e, então, ao encontrar uma música, poder ouvi-la por completo. Da mesma forma para o SoundHound, além de ser possível o compartilhamento da sua pesquisa com o Twitter. Com o Musixmatch integrado ao Spotify, por exemplo, ao encontrar uma música, você acompanha a música com a letra em tempo real. O ACRCLOUD e o Musixmatch são as soluções que possuem o maior número de integrações.

Com relação à usabilidade mostrado na Tabela 6, testes foram realizados considerando os critérios e subcritérios descritos na seção 4.1. Foi realizada uma adaptação do *MATcH Checklist* disponibilizado pelo Grupo de Qualidade de Software (GQS) da UFSC, onde o conjunto de perguntas possui uma escala de resposta com 2 opções: Sim (a solução atende o objetivo) e Não (a solução não atende o objetivo). Cada solução possui particularidades que podem atrair ou afastar o usuário. Os resultados dos testes são discriminados abaixo, feitos em um smartphone, modelo Asus Zenfone 4, com sistema operacional Android.

Todas as soluções são simples, possuem navegação intuitiva e são de fácil execução. As funções mais utilizadas são facilmente acessadas. Os componentes interativos são claramente distintos uns dos outros, com ícones intuitivos. Com exceção do Musipedia, as soluções possuem uma linguagem clara, concisa e funcionam corretamente, não apresentando problemas. Com relação à Musipedia (ver Figura 32), ela é uma enciclopédia de músicas e sua navegação não é muito intuitiva, além de possuir textos longos, não apresentando uma linguagem clara e concisa. Não funciona corretamente, não apresentando mensagens de erros claras e objetivas.

No caso do SoundHound, Deezer e Spotify, o menu é esteticamente simples, mas suas versões gratuitas possuem propagandas, o que polui a tela. Ao se adquirir a versão paga, as propagandas são retiradas. Já o MusicID e o Musixmatch não possuem muitas funcionalidades e o menu é esteticamente simples, claro e sem propagandas.

Deezer, Spotify e SoundCloud executam buscas de músicas pela correspondência de texto (metadados) para identificar arquivos de música digital, ou seja, através da informação do título, álbum ou gênero da música.

O MusicID (ver Figura 24) é de uso exclusivo para o reconhecimento de músicas ouvindo você cantar, cantarolar ou que são tocadas ao seu redor. O mesmo vale para o SoundHound (ver Figura 26), com o diferencial de possuir mais funcionalidades além do reconhecimento de músicas, como a possibilidade de criar playlists e o compartilhamento



com o Twitter. Além das informações básicas sobre a música, o aplicativo sugere alguns vídeos que podem ser assistidos diretamente no SoundHound.

O Shazam (ver Figura 25) é similar ao MusicID e ao SoundHound. A diferença é que faz o reconhecimento de músicas apenas tocadas ao seu redor. Ele também possui a funcionalidade de tirar fotos de QRCodes para realizar a busca da música. O Deezer (ver Figura 27) é similar ao Shazam, pois também possui a funcionalidade de reconhecimento de músicas que são tocadas ao seu redor e a possibilidade do download das músicas para uso off-line, além da criação de playlists.

O Spotify (ver Figura 28) é similar ao Deezer em termos de funcionalidades. A diferença é que o Spotify não possui reconhecimento das músicas pelo som (trecho de áudio ou voz). Para usufruir de todas as possibilidades do aplicativo, como o download de músicas para uso off-line, é necessário obter a versão premium. Por sua vez, o SoundCloud (ver Figura 29) é similar ao Spotify, com a diferença de ser o único aplicativo comercial analisado que permite a inclusão de músicas criadas pelos usuários.

O Musixmatch (ver Figura 30) é um aplicativo que sincroniza letras de músicas e também permite o reconhecimento de músicas que são tocadas ao seu redor. Ainda, o ACRCLOUD (ver Figura 31) é um serviço na nuvem, com um grande banco de dados musical. Além de oferecer o serviço de reconhecimento de músicas, ele também permite o reconhecimento de músicas pela web, como um teste do seu serviço.

#### **4.2.2 Soluções Acadêmicas**

Em relação ao critério de Eficiência mostrado na Tabela 7, o comportamento das soluções em relação ao tempo foi encontrada documentação apenas para o Tunebot. A solução utiliza QBH e, segundo Lab (2008b), o comprimento mediano de uma música cantada por um usuário é de cerca de 18 segundos de áudio, levando cerca de 5s para retornar resultados a partir do momento em que a aplicação recebe a música. Para comparação, a música mais longa recebida até o momento tem cerca de 48 segundos e a aplicação demora cerca de 13s para retornar uma resposta. O tempo de retorno é uma função de vários fatores, incluindo o tamanho do banco de dados e o comprimento da consulta, tanto em termos da duração total do áudio quanto do número de notas que o usuário cantou.

Cada solução possui ferramentas variadas para uso distinto no reconhecimento de músicas, o que gera diversas formas de utilização dos recursos. Por exemplo, o MusicMiner (ver Figura 37) possui ferramentas que utilizam os métodos de Classificação (ver subseção 2.5.1.2) e Visualização (ver subseção 2.5.1.11) através de Mapas Musicais Topográficos.

O AMUSE (ver Figura 33) e o Java MIR (ver Figura 35) utilizam o método de Classificação (ver subseção 2.5.1.2). Já o Tunebot (ver Figura 39 e 38) utiliza Query by Humming (QBH) (ver subseção 2.5.1.8). O CLAM (ver Figura 34) utiliza Spectral Modeling Synthesis (SMS), e o MIRtoolbox (ver Figura 36) é um acoplado de ferramentas baseado em técnicas de mineração de dados. Por ser um pacote de ferramentas, são variadas técnicas para recuperação da informação musical que ele oferece, como por exemplo, clusterização e classificação.

Quanto ao critério de bitrate, das 6 soluções analisadas, não foi encontrada documentação que especificasse o bitrate utilizado para o fluxo de transferência das músicas.

Em relação ao critério de Adequação Funcional mostrado na Tabela 8, as soluções acadêmicas apresentam um modelo de desenvolvimento aberto, ou seja, seu código pode ser alterado por outros usuários, contribuindo para a resolução de problemas existentes nas aplicações. Das 6 soluções analisadas, todas possuem reconhecimento de músicas de forma aproximada (A), ou seja, permitem realizar buscas de músicas através de trechos de áudio e/ou voz e/ou codificação MIDI e/ou por conteúdo (metadados).

Todas as soluções permitem a inclusão de músicas criadas por usuários. Quanto à disponibilidade, com exceção do Tunebot, todas as soluções estão disponíveis para Linux, Mac OSX e Windows e seu acesso é off-line, ou seja, é necessário o download e instalação do projeto no dispositivo. O download e instalação das soluções é feito de forma gratuita. O Tunebot está disponível na Web e também possui uma versão beta para smartphones iOS, tendo o seu acesso on-line (Ver Figura 38).

O AMUSE e o Tunebot são as duas soluções que possuem integração com outras aplicações. O AMUSE faz parte do rol de ferramentas do MIRtoolbox. Já o Tunebot tem integração com o *Karaoke Callout*<sup>1</sup>, que ajuda a construir a base de conhecimento de músicas do Tunebot.

Como explicitado anteriormente, as soluções analisadas tem seu acesso off-line. Diante disso, para o funcionamento correto de cada aplicação é necessário a instalação de projetos de terceiros. Desta forma, com a impossibilidade de funcionamento das aplicações, não foi possível analisar os critérios de usabilidade das soluções acadêmicas.

#### 4.3 DISCUSSÃO DOS RESULTADOS

Conforme explicado na seção anterior, a análise de soluções foi dividida em dois grupos: soluções comerciais e soluções acadêmicas. Uma análise entre todas elas seria

---

<sup>1</sup>*Karaoke Callout* é um jogo de karaokê para a plataforma iOS da Apple que permite aos usuários "tunebot"desafiarem uns aos outros para um duelo de canto (SHAMMA; PARDO, 2006).

inviável, pois cada grupo se difere no motivo de uso e é voltado a públicos-alvo diferentes. As soluções comerciais são mais utilizadas por usuários finais. Já as soluções acadêmicas são voltadas a usuários pesquisadores da comunidade de MIR.

#### 4.3.1 Soluções Comerciais

Com relação à análise das soluções comerciais, pode-se concluir que as soluções focam na usabilidade visando facilitar o uso para o seu público-alvo, que apenas as utilizam no seu dia-a-dia. Estes usuários não possuem muito interesse em como essas soluções funcionam, desejando apenas que a busca seja rápida e satisfaça os filtros informados.

Os testes realizados de reconhecimento e busca de músicas, por metadados ou por som, demonstraram que as aplicações atenderam bem a todas as funcionalidades oferecidas. Se o objetivo for o reconhecimento de músicas quando não se conhece nenhum valor de metadado, o Shazam e o Deezer são essenciais. Porém, se o objetivo for aquela música que não sai da cabeça e precisar cantar ou cantarolar, então o ideal é usar o SoundHound ou o MusicID. Se o conhecimento da letra da música for indispensável, é possível usar o Musixmatch.

Se o objetivo for playlists e download de músicas, para não ter a necessidade do uso de conexão com a Internet, o Spotify ou o Deezer atendem a demanda. Já o Musipedia, em particular, é uma solução voltada a usuários músicos, que compõem músicas através da codificação MIDI.

Em relação ao tempo, o ideal é a conexão de Internet Wi-Fi. Nestas condições, todas as soluções, exceto o Musipedia, fazem o reconhecimento/busca de músicas no tempo mínimo de 5s. Nesse caso, saber o objetivo do usuário com relação a desempenho é imprescindível para a escolha de qual solução usar.

Pode-se concluir que o método mais utilizado para o reconhecimento de músicas é o *Fingerprint*. Inclusive, o uso do serviço do ACRCLOUD tem sido incorporado em cada vez mais soluções para o reconhecimento das músicas, como é o caso do Musixmatch e do Deezer. Outro método bastante utilizado é o RPC, principalmente pelas soluções que permitem apenas buscas exatas, onde é feita a correspondência de texto para identificar arquivos de música digital, ou seja, através da informação do título, álbum ou gênero da música.

### 4.3.2 Soluções Acadêmicas

Com relação à análise realizada para as soluções acadêmicas, pode-se concluir que as soluções focam na otimização e desempenho no que diz respeito ao reconhecimento de músicas. O desenvolvimento de tais aplicações é voltado para os usuários pesquisadores da comunidade de MIR, com o objetivo de trazer inovação na busca de dados musicais.

Todas as soluções incluem uma estratégia para busca por similaridade (a maioria delas utilizando técnicas de Classificação), são gratuitas e permitem a inclusão de dados musicais em seus repositórios. Ainda, a maioria são soluções off-line e estão disponíveis em diversas plataformas. Por outro lado, poucas soluções oferecem integração com outras soluções ou aplicações que tocam músicas. Inclusive apresentam um modelo de desenvolvimento aberto, ou seja, seu código pode ser alterado por outros usuários.

Houve dificuldade no uso das soluções acadêmicas e por este motivo, não houve a análise dos critérios de usabilidade, além de não ter sido possível o teste prático dos critérios de eficiência de desempenho, o que poderia ter enriquecido ainda mais a presente pesquisa. A documentação existente para cada proposta sugere que o desenvolvimento das soluções acadêmicas geralmente tem o objetivo de validar uma determinada pesquisa, sendo após oferecida gratuitamente à comunidade para futuras pesquisas e o aprimoramento da solução. Desta forma, a usabilidade nem sempre é o foco principal.

Por outro lado, certos processos já foram desenvolvidos e estão sendo utilizados como parte de uma nova solução, como é o caso do projeto AMUSE, que faz parte do rol de ferramentas do MIRtoolbox. Outro caso é o Tunebot, que possui uma versão beta desenvolvida para plataformas iOS, sendo um dos seus objetivos ajudar pesquisadores na área de reconhecimento de músicas que utilizam o algoritmo *Query by Humming*, facilitando uma pesquisa mais precisa do desempenho do mundo real do que seria possível com conjuntos de dados existentes.

Por fim, a escolha de um método para a recuperação da informação musical vai depender do objetivo da solução a ser desenvolvida. Neste sentido, quanto mais aprimorados forem os métodos e algoritmos de soluções acadêmicas, melhores soluções comerciais serão desenvolvidas para a utilização pelo usuário final.

Tabela 5 – Análise das soluções comerciais: Critério de Adequação Funcional

CRITÉRIOS		Adequação funcional						
		Disponibilidade	Modelo de desenv.	Integrações	Acessibilidade	Busca de dados	Inclusão de dados	Modelo de Pagamento
C O M P R E C I S A L	<b>MusicID</b>	iOS, Android	F	não	On	A	não	G
	<b>Shazam</b>	iOS, Android	F	Spotify, Google Music, Apple Music	On	E	não	G
	<b>SoundHound</b>	iOS, Android	F	Spotify, Youtube, Twitter	On	A	não	F
	<b>Deezer</b>	iOS, Android, Windows, Web	F	Google Music, Twitter, Facebook	On/Off	E	não	F
	<b>Spotify</b>	iOS, Android, Windows, Web Linux, OS X	F	não	On/Off	E	não	F
	<b>SoundCloud</b>	iOS, Android	F	não	On	E	sim	F
	<b>Musixmatch</b>	iOS, Android	F	Spotify, Deezer, Google Music, Youtube, Twitter, Facebook, Last.fm	On	E	não	F
	<b>ACRCloud</b>	Web	F	Spotify, Deezer, Youtube, ISRC, UPC, LyricFind, Music Story, SyncPower	On	A	sim	P
	<b>Musipedia</b>	Web	A	não	On	A	sim	G

Legenda: F - Fechado; A - Aberto; E - Busca Exata; A - Busca Aproximada; G - Gratuito; F - Freemium; P - Premium;

Tabela 6 – Análise das soluções comerciais: Critério de Usabilidade

CRITÉRIOS		Usabilidade				
		Visibilidade do status do sistema	Prevenção de erros	Flexibilidade e eficiência de uso	Estética e Design minimalista	Pouca interação homem/dispositivo
C	<b>MusicID</b>	sim	sim	sim	sim	sim
O	<b>Shazam</b>	sim	sim	sim	sim	sim
M	<b>SoundHound</b>	sim	sim	sim	sim	sim
E	<b>Deezer</b>	sim	sim	sim	sim	sim
R	<b>Spotify</b>	sim	sim	sim	sim	sim
C	<b>SoundCloud</b>	sim	sim	sim	sim	sim
I	<b>Musixmatch</b>	sim	sim	sim	sim	sim
A	<b>ACRCloud</b>	sim	sim	sim	sim	sim
L	<b>Musipedia</b>	sim	não	sim	não	sim

Tabela 7 – Análise das soluções acadêmicas: Critério de Eficiência

CRITÉRIOS		Eficiência		
		Comportamento em relação ao tempo	Utilização de recursos	Bitrate
A	<b>MusicMiner</b>	-	C e V	-
C	<b>CLAM</b>	-	SMS	-
A	<b>MIRtoolbox</b>	-	Diversas	-
D	<b>AMUSE</b>	-	C	-
Ê	<b>Java MIR</b>	-	C	-
	<b>Tunebot</b>	Até 13s	QBH	-

Legenda: C - Classificação; V - Visualização; SMS - Spectral Modeling Synthesis; QBH - Query by Humming;

Tabela 8 – Análise das soluções acadêmicas: Critério de Adequação Funcional

CRITÉRIOS		Adequação funcional						
		Disponibilidade	Modelo de desenv.	Integrações	Acessibilidade	Busca de dados	Inclusão de dados	Modelo de Pagamento
A C A D Ê M I C O	<b>MusicMiner</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>CLAM</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>MIRtoolbox</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>AMUSE</b>	-	A	MIRtoolbox	Off	A	sim	G
	<b>Java MIR</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>Tunebot</b>	Web, iOS	A	Karaokê Callout	On	A	sim	G

Legenda: F - Fechado; A - Aberto; E - Busca Exata; A - Busca Aproximada; G - Gratuito; F - Freemiun; P - Premium;

## 5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo analisar soluções para a busca por similaridade de dados musicais, ou seja, soluções que não necessariamente buscam dados musicais apenas através do casamento direto de parâmetros de entrada para a busca, como título da música, palavras-chave ou um áudio como parte da música, mas também através do casamento aproximado (ou similar) destes parâmetros. Além de uma pesquisa aplicada sobre o que são dados musicais, sobre a forma como os dados são tratados e armazenados, para que possam posteriormente ser recuperados e ouvidos no dia-a-dia, este trabalho buscou reunir e comparar informações com o propósito de contribuir com futuros trabalhos que desejam desenvolver soluções para a busca por similaridade de dados musicais.

Pela observação das características analisadas nas Tabelas 4, 5 e 6, as soluções comerciais enfatizam a usabilidade, visando facilitar o uso para o público-alvo, que é o usuário final no seu dia-a-dia. O objetivo destas soluções pode ser o reconhecimento de músicas através de uma parte da música ou da voz, bem como a criação de playlists e download de músicas para uso off-line. O método mais utilizado para o reconhecimento de músicas é o Fingerprint.

As soluções acadêmicas analisadas nas Tabelas 7 e 8, por sua vez, enfatizam o desempenho no reconhecimento de músicas. O desenvolvimento de tais soluções é destinado principalmente a usuários pesquisadores da comunidade de MIR com o objetivo de trazer inovação na maneira como são recuperados os dados musicais. Todas as propostas incluem uma estratégia para busca por similaridade (a maioria delas utilizando técnicas de Classificação), são gratuitas e permitem a inclusão de dados musicais em seus repositórios. Ainda, a maioria são soluções off-line e estão disponíveis em diversas plataformas. Por outro lado, poucas soluções oferecem integração com outras soluções ou aplicações que tocam músicas.

Como trabalhos futuros, sugere-se, primeiramente, a aquisição dos protótipos das soluções acadêmicas para que uma avaliação de usabilidade destas propostas possa ser realizada. Neste trabalho, devido a restrições de tempo, não foi possível ter acesso a essas soluções. Outra sugestão é um aprofundamento no estudo dos métodos e algoritmos utilizados para a recuperação da informação musical, como *FingerPrint*, Recuperação por Conteúdo e Classificação, bem como uma comparação para verificar os métodos mais eficientes. Por fim, o desenvolvimento e avaliação de soluções para recuperação de informação musical poderia contar com a participação de mais mulheres no programa de mentoria



WIMIR<sup>1</sup>, incluindo soluções para busca por similaridade.

---

<sup>1</sup><https://wimir.wordpress.com/mentoring-program/>

## REFERÊNCIAS

ACRCLLOUD. *ACRCloud Serviços de Reconhecimento Automático e Confiável na nuvem ACRCloud*. 2015. <<https://www.acrcloud.com/docs/acrcloud/>>.

ACRCLLOUD. *ACRCloud Serviços de Reconhecimento Automático e Confiável na nuvem ACRCloud*. 2015. <<https://www.acrcloud.com/identify-songs-music-recognition-online/>>.

ALLIGOOD, K. T.; SAUER, T. D.; YORKE., J. A. Chaos: An introduction to dynamical systems. *Research Triangle Park*, v. 85, p. 603 p., oct 1996.

ALMEIDA, M. C. B. de. *A informação nas aéreas de arte: um olhar além das práticas*. Recife/PE, 2007. 253 – 272 p.  
<<http://www2.eca.usp.br/nucleos/colabori/documentos/Infoeducacao.pdf>>.

AMATRIAIN, X. *An Object-Oriented Metamodel for Digital Signal Processing with a focus on Audio and Music*. Tese (PhD thesis) — universitat Pompeu Fabra, Barcelona, 2004.

AMATRIAIN, X. Clam: A framework for audio and music application development. *IEEE Softw* 24(1), p. 82–85, 2007.

ANDRADE, V. G. D.; CRISPIM, E. A.; OGIBOWSKI, A. R. G. *MGM – Modelo de Gestão Musical*. 78 p. Dissertação (Graduação) — Universidade Católica de Brasília, Brasília, 2008.

ANGELES, B.; MCKAY, C.; FUJINAGA, I. Discovering metadata inconsistencies. In: ANAIS ELETRÔNICOS. *International Society for Music Information Retrieval Conference*. Utrecht (Holanda), 2010.  
<[http://jmir.sourceforge.net/publications/ISMIR\\_2010\\_Discovering.pdf](http://jmir.sourceforge.net/publications/ISMIR_2010_Discovering.pdf)>.

BARIONI, M. C. N. *Operações de consulta por similaridade em grandes bases de dados complexos*. 170 p. Tese (Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação - ICMC-USP, São Carlos, 2006.

BARROS, C. M. de. *Representação da Informação Musical: Subsidios para Recuperação da Informação em Registros Sonoros e Partituras no Contexto Educacional e de Pesquisa*. 150 p. Dissertação (Pós-Graduação) — Universidade Federal de Santa Catarina-UFSC, Florianópolis, 2012.

BERGER, P. L.; LUCKMANN, T. *A construção social da realidade*. 36. ed. Petrópolis: Editora Vozes, 2014. ISBN 9788532605986.

BOGORNY, V. *Classificação*. 2018.  
<[http://www.inf.ufsc.br/vania/teaching/INE5644/classificacao\\_arvores.ppt](http://www.inf.ufsc.br/vania/teaching/INE5644/classificacao_arvores.ppt)>.

BOGORNY, V. *Clustering*. 2018.  
<<http://www.inf.ufsc.br/vania/teaching/INE5644/clustering1.ppt>>.

BOGORNY, V. *Clustering*. 2018.

<<http://www.inf.ufsc.br/vania/teaching/INE5644/clustering2.ppt>>.

BOHM, C.; BERCHTOLD, S.; KEIM, D. A. Searching in high-dimensional spaces - index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, p. 322–373, 2001.

BONADA, J. et al. Spectral approach to the modeling of the singing voice. In: *In Proc. of the 111th AES Convention*. [S.l.: s.n.], 2001.

BOZKAYA, T.; ÖZSOYOGLU, M. Distance-based indexing for high-dimensional metric spaces. In: ANAIS ELETRÔNICOS. *23rd ACM International Conference on Management of Data (SIGMOD)*. Tucson, USA, 1997. p. 357–368.

BRIN, S. Near neighbor search in large metric spaces. In: ANAIS ELETRÔNICOS. *21st International Conference on Very Large Data Bases (VLDB)*. Zurich, Switzerland, 1995. p. 574–584.

BUNNELL, T. *Sound in the frequency domain*. feb 1996a.

<[http://www.asel.udel.edu/speech/tutorials/acoustics/freq\\_domain.html](http://www.asel.udel.edu/speech/tutorials/acoustics/freq_domain.html)>.

CANO, P. et al. A review of audio fingerprinting. *Journal of VLSI Signal Processing*, v. 41, p. 271–284, 2005.

CARREIRA, E. B. *Aplicativo Web para Análise e Comparação de Áudio Fingerprint*. Dissertação (Graduação) — Universidade Estadual Paulista Júlio de Mesquita Filho, Faculdade de Ciências, Campus Bauru, Bauru, 2015.

CARTWRIGHT, M.; PARDO, B. Building a music search database using human computation. *Proceedings of the 9th Sound and Music Computing Conference (SMC 2012)*, Copenhagen, Denmark, July 12-14, 2012, jul 2012.

CHÁVEZ, E. et al. Searching in metric spaces. *ACM Computing Surveys*, CRC Press, p. 273–321, 2001.

CIACCIA, P.; PATELLA, M.; ZEZULA, P. M-tree: An efficient access method for similarity search in metric spaces. In: ANAIS ELETRÔNICOS. *23rd International Conference on Very Large Data Bases (VLDB)*. Athens, Greece, 1997. p. 426–435.

CORMEN, T. H. et al. *Introduction to Algorithms, Third Edition*. 3rd. ed. [S.l.]: The MIT Press, 2009. ISBN 0262033844, 9780262033848.

CRUZ, L. F. *Um Método para Recuperação de Composições Polifônicas aplicado na Busca de Tablaturas Textuais da Web*. 79 p. Dissertação (Graduação) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

DAQUINO, F. *A evolução do armazenamento de músicas [infográfico]*. Outubro 2012. <<https://www.tecmundo.com.br/infografico/30658-a-evolucao-do-armazenamento-de-musicas-infografico-.htm>>.

DEEZER. *Deezer*. 2006. <<https://www.deezer.com/br/company>>.

DEVELOPMENT, C. *C++ Library for Audio and Music - CLAM*. 2004. <<http://clam-project.org/>>.

DEVELOPMENT, M. *Musipedia The Open Music Encyclopedia*. 2018? <<https://www.musipedia.org/about.html>>.

DORTMUND, T. *AMUSE - Advanced Music Exploration*. Germany: [s.n.], 2018? <<http://amuse-framework.sourceforge.net>>.

DOWNIE, J. S. *Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text*. Tese (Pós-Graduação) — University of Western Ontario, London, Ontário, Canadá, 1999.

DOWNIE, J. S. Music information retrieval (chapter 7). In: BLAISE CRONIN. *Annual Review of Information Science and Technology*. Medford, NJ: Information Today, 2003. p. 295 – 340. <[http://music-ir.org/downie\\_mir\\_arist37.pdf](http://music-ir.org/downie_mir_arist37.pdf)>.

ECKMANN, J.-P.; KAMPHORST, S. O.; RUELLE, D. Recurrence plots of dynamical systems. *Europhys. Lett.* 4 973, v. 16, p. 973–977, nov 1987.

ENESCO, M.; DHANDHANIA, K.; SVETLECIC, D. *K-Means Clustering*. 2018? <<https://www.commonlounge.com/discussion/665476f64e574b0fa259a15423ba69cc>>.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. *proc. kdd*. 1996.

F, S. *iTunes's new 1080p Compares Well to Blu-ray Quality, At a Fraction of File Size*. 2012. <<http://www.digital-digest.com/news-63300-iTunes-new-1080p-Compares-Well-to-Blu-ray-Quality-At-a-Fraction-of-File-Size.html>>.

FERREIRA, J. B. *Recuperação de Informação de Música e dados ID3: possíveis aplicações*. Dissertação (Dissertação (Mestrado em Ciência da Informação)) — Universidade Estadual Paulista, Faculdade de Filosofia e Ciências, Marília, 2015.

FILHO, R. F. S. et al. Similarity search without tears: The omni family of all-purpose access methods. In: *17th IEEE International Conference on Data Engineering (ICDE)*. Heidelberg, Germany: [s.n.], 2001. p. 623–630.

FUTRELLE, J.; DOWNIE, J. S. Interdisciplinary communities and research issues in music information retrieval. In: ANAIS ELETRÔNICOS. *The International Society of Music Information Retrieval*. Champaign, Illinois, 2002. p. 1 – 7. <<http://ismir2002.ismir.net/proceedings/02-FP07-3.pdf>>.

GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, J. *Database systems: The complete book*. Prentice Hall, Upper Saddle River, New Jersey: Editora Ciência Moderna, 2002.

GHIAS, A. et al. Query by humming: Musical information retrieval in an audio database. In: *Proceedings of the Third ACM International Conference on Multimedia*. New York, NY, USA: ACM, 1995. (MULTIMEDIA '95), p. 231–236. ISBN 0-89791-751-0. <<http://doi.acm.org/10.1145/217279.215273>>.

GOMES, A. C. F. *Representação da informação musical: uma análise a luz dos Requisitos Funcionais para Dados de Autoridade (FRAD)*. 101 p. Dissertação (Graduação) — Universidade de Brasília, Brasília, 2015.

GRACENOTE. *MusicID*. 1998. <<http://www.gracenote.com/music/music-recognition/>>.

GUTIÉRREZ, E. G. *TONAL DESCRIPTION OF MUSIC AUDIO SIGNALS*. Tese (Doutorado) — Universitat Pompeu Fabra for the Program in Computer Science and Digital, Barcelona, 2006.

HAITSMA, J.; KALKER, T. A highly robust audio fingerprinting system. In: ANAIS ELETRÔNICOS. *International Society for Music Information Retrieval Conference*. [S.l.], 2002.

HENRIQUES, M. *Análise Semântica de Sinais Audio: Audio Fingerprinting*. [S.l.], may 2003.

HUQ, A.; CARTWRIGHT, M.; PARDO, B. Crowdsourcing a real - world on - line query by humming system. *Proceedings of the 7th Sound and Music Computing Conference (SMC 2010), Barcelona, Spain, July 21-24, 2010*, jul 2010.

HUQ, A.; CARTWRIGHT, M.; PARDO, B. Crowdsourcing a real-world on-line query by humming system. In: *Proceedings of the SMC 2010 - 7th Sound and Music Computing Conference*. Barcelona, Espanha: [s.n.], 2010.

IAZZETTA, F.; KON, F. *A Música Efêmera da Internet*. Dissertação (Mestrado) — Comunicação e Semiótica - PUC-SP, University of Illinois at Urbana-Champaign, 1998.

JUANG, B. H.; RABINER, L. R. Hidden markov models for speech recognition. *Technometrics*, v. 33, n. 3, p. 251–272, aug 1991.

JUNIOR, E.; SEGUNDO, A. *Histórico dos Banco de Dados*. Março 2008. <[disciplinas.decc.ufba.br/svn/MATA60/tarefa1/historico/historico.pdf%3Frevision%3D21](http://disciplinas.decc.ufba.br/svn/MATA60/tarefa1/historico/historico.pdf%3Frevision%3D21)>.

JYVÄSKYLÄ, U. of. *MIRtoolbox*. Finlândia: [s.n.], 2018? <<https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox>>.

KAISTER, D. dos S. *Tratamento de condições especiais para busca por similaridade em bancos de dados complexos*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação - ICMC-USP, 2012. 177 p.

KEOGH, E.; RATANAMAHATANA, C. A. Exact indexing of dynamic time warping. *Knowledge and Information Systems 7.3*, v. 7, n. 10.1007/s10115-004-0154-9, 2004.

KRUSKAL, J.; LIBERMAN, M. The symmetric time-warping problem: From continuous to discrete. 01 1983.

LAB, I. A. *Tunebot*. 2008. <<http://music.eecs.northwestern.edu/research.php?project=tunebot>>.

LAB, I. A. *Tunebot Dataset*. 2008. <<http://music.cs.northwestern.edu/data/tunebot/>>.

- LARTILLOT, O. *MIRtoolbox User's Guide 1.7.1*. 2013.  
<<https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox/manual1-7-1.pdf>>.
- LARTILLOT, O.; TOIVIAINEN, P. Mir in matlab (ii): A toolbox for musical feature extraction from audio. In: *Proc. 8th International Conference on Music Information Retrieval (ISMIR 2007)*. Vienna: [s.n.], 2007. p. 127–130.
- LIMA, C. R. M. de; SANTINI, R. M. O uso do creative commons no compartilhamento de música na internet. In: *VII Encontro Nacional de Pesquisa em Ciência da Informação - ENANCIB*. Marília: [s.n.], 2006.  
<[http://www2.eptic.com.br/ulepica\\_brasil/arquivos/ec\\_limasantini.pdf](http://www2.eptic.com.br/ulepica_brasil/arquivos/ec_limasantini.pdf)>.
- MARBURG, U. of. *MusicMiner - Databionic MusicMiner*. Germany: [s.n.], 2005.  
<<http://musicminer.sourceforge.net/>>.
- MARCHI, L. D. A angústia do formato: uma história dos formatos fonográficos. *Revista E-Compos*, Revista da Associação Nacional dos Programas de Pós-Graduação em Comunicação, Rio de Janeiro, p. 1 – 19, 2005. <<http://www.e-compos.org.br/e-compos/article/viewFile/29/30>>.
- MCKAY, C. *Automatic music classification with jMIR*. Tese (PhD thesis) — CIRMMT and Marianopolis College, Québec, 2010.
- MCKAY, C. *jMIR*. CIRMMT and Marianopolis College: [s.n.], 2018?  
<<http://jmir.sourceforge.net/>>.
- MCLANE, A. Music as information. In: AMERICAN SOCIETY FOR INFORMATION SCIENCE. *Annual Review of Information Science and Technology (ARIST)*. [S.l.], 1996. v. 31, p. 225 – 262.
- MESAROS, A. Singing voice identification and lyrics transcription for music information retrieval invited paper. In: *2013 7th Conference on Speech Technology and Human - Computer Dialogue (SpeD)*. [S.l.: s.n.], 2013. p. 1–10.
- MICHELS, U. *Atlas de música*. 2. ed. Madrid: Alianza, 1992.
- MILETTO, E. M. et al. Minicurso: Introdução a computação musical. In: ANAIS ELETRÔNICOS. *CBComp - Congresso Brasileiro de Computação*. Itajaí, 2004. p. 883 – 902. ISSN 1677-2822.  
<[https://musicaeadoracao.com.br/recursos/arquivos/tecnicos/outros/computacao\\_musical.pdf](https://musicaeadoracao.com.br/recursos/arquivos/tecnicos/outros/computacao_musical.pdf)>.
- MIZUTANI, E. The dynamic time warping algorithms. *Lecture Note for Mechanical Engineering Seminar, Tokyo Metropolitan University*, 2006.
- MORCHEN, F. et al. Databionic visualization of music collections according to perceptual distance. In: *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*. London, UK: [s.n.], 2005. p. 396–403.
- MULLER, M. *Information Retrieval for Music and Motion*. Bonn, Germany: Springer Berlin Heidelberg New York, 2007. (ISBN 978-3-540-74047-6).

- MURTHY, Y. V. S.; KOOLAGUDI, S. G. Content-based music information retrieval (cb-mir) and its applications toward the music industry: A review. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 51, n. 3, p. 45:1–45:46, Jul 2018. ISSN 0360-0300. <<http://doi.acm.org/10.1145/3177849>>.
- MUSIXMATCH. *Musixmatch*. 2010. <<http://about.musixmatch.com/>>.
- NIELSEN, J. 10 usability heuristics for user interface design. 1995. <<https://www.nngroup.com/articles/ten-usability-heuristics/>>.
- PACHECO, J.; LOPES, R. Descrição e procura de música. *Comunicação Áudio e Vídeo, Instituto Superior Técnico*, 2014.
- PINHEIRO, L. V. R.; LOUREITO, J. M. M. Traçados e limites da ciência da informação. *Ciencia da Informação*, Brasília, v. 24, n. 1, p. 42 – 53, 1995. <<http://revista.ibict.br/index.php/ciinf/article/view/531/483>>.
- REZENDE, J. L. de et al. *Orfeu, um sistema para busca e manipulação de arquivos de música*. Dissertação (Mestrado) — Coordenação dos Programas de Pós-Graduação de Engenharia da Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Ilha do Fundão – Rio de Janeiro – RJ – Brazil, 2004.
- RYYNÄNEN, M.; KLAPURI, A. Query by humming of midi and audio using locality sensitive hashing. *IEEE Xplore Digital Library*, p. 2249–2252, 2008.
- SALAMON, J.; ROHRMEIER, M. A quantitative evaluation of a two stage retrieval approach for a melodic query by example system. In: *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*. Kobe, Japan: [s.n.], 2009. <<files/publications/JSalamon-ismir09.pdf>>.
- SANDISK. *Definição de Bitrate*. 2018? <[https://kb-pt.sandisk.com/app/answers/detail/a\\_id/14843/ /definição-de-bitrate](https://kb-pt.sandisk.com/app/answers/detail/a_id/14843/ /definição-de-bitrate)>.
- SANTINI, R. M.; SOUZA, R. F. de. Recuperação da informação de música e a ciência da informação: Tendências e desafios de pesquisa. In: ORGANIZAÇÃO E REPRESENTAÇÃO DO CONHECIMENTO POSTER. *VIII ENANCIB – Encontro Nacional de Pesquisa em Ciência da Informação*. Salvador, Bahia, 2007. p. 1 – 14. <<http://www.enancib.ppgci.ufba.br/artigos/GT2-143.pdf>>.
- SANTOS, E. J. de C. A. *Ferramentas de Busca Musical Baseada em Reconhecimento de Melodias: Estudo, Análise e Implementação*. Dissertação (Graduação) — Universidade de Pernambuco, Recife - PE, may 2011.
- SEMIDÃO, R. A. M. Dados, informação e conhecimento: elementos de análise conceitual. *DataGramaZero*, v. 14, n. 4, aug 2013.
- SERRA, J. e X.; ANDRZEJAK, R. G. Cross recurrence quantification for cover song identification. *New Journal of Physics* 11.9, v. 11 093017, n. 0.1088/1367-2630/11/9/093017, Setembro 2009.
- SERRA, X.; SMITH, J. Spectral modeling synthesis: A sound analysis/synthesis based on a deterministic plus stochastic decomposition. *Computer Music Journal*, v. 14, p. 12–24, 1990. SMS. <<http://hdl.handle.net/10230/33796>>.

SETZER, V. W. Dado, informação, conhecimento e competência. *Datagrama*, Sao Paulo, v. 10, p. 1 – 14, 2001. <<http://www.ime.usp.br/%7Evwsetzer/dado-info.html>>.

SHAMMA, D.; PARDO, B. Karaoke callout: using social and collaborative cell phone networking for new entertainment modalities and data collection. In: *In Proceedings of ACM Multimedia Workshop on Audio and Music Computing for Multimedia (AMCMM 2006)*. Santa Barbara, CA, USA: [s.n.], 2006. <<http://music.eecs.northwestern.edu/publications/shamma-pardo-acmmm-06.pdf>>.

SHAZAM. *Shazam*. 2000. <<https://www.shazam.com/pt/company>>.

SHERIDAN, K. *O Som do Coração*. 2008. <<http://www.adorocinema.com/filmes/filme-109845/creditos/>>.

SILVA, E. L. da; MENEZES, E. M. *Metodologia da Pesquisa e Elaboração de Dissertação*. 138 p. Tese (Pós-Graduação) — UFSC, Florianópolis, 2005. 4. ed. rev. atual. <[https://projetos.inf.ufsc.br/arquivos/Metodologia\\_de\\_pesquisa\\_e\\_elaboracao\\_de\\_teses\\_e\\_diss](https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_diss)>.

SILVA, L. M. O. da. *Uma Aplicação de Árvores de Decisão, Redes Neurais e KNN para a Identificação de Modelos ARMA Não-Sazonais e Sazonais*. Dissertação (Pós-graduação) — PUC-Rio, Rio de Janeiro - RJ, set 2005. <[https://www.maxwell.vrac.puc-rio.br/7587/7587\\_6.PDF](https://www.maxwell.vrac.puc-rio.br/7587/7587_6.PDF)>.

SOUNDCLOUD. *SoundCloud*. 2007. <<https://soundcloud.com/pages/contact>>.

SOUNDHOUND. *SoundHound Music Search & Play*. 2005. <<https://soundhound.com/about>>.

SPOTIFY. *Spotify*. 2006. <<https://www.spotify.com/br/>>.

SUZUKI, M. et al. Music information retrieval from a singing voice using lyrics and melody information. *EURASIP Journal on Advances in Signal Processing*, v. 2007, n. 38727, p. 8, sep 2006.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introdução ao Data Mining Mineração de Dados*. Rio de Janeiro: Editora Ciência Moderna, 2009.

TRAINA, A. J. M.; TRAINA-JR., C. Similarity search in multimedia databases. *Handbook of video databases: design and applications*, CRC Press, p. 711–738, 2003.

TRAINA, C. J. et al. Slim-trees: High performance metric trees minimizing overlap between nodes. In: *7th IEEE International Conference on Extending Database Technology (EDBT)*. Konstanz, Germany: [s.n.], 2000. p. 51–65.

TYPKE, R. et al. Using transportation distances for measuring melodic similarity. In: *ISMIR Proceedings*. [S.l.: s.n.], 2003. p. 107–114.

TZANETAKIS, G.; ERMOLINSKIY, A.; COOK, P. R. Pitch histograms in audio and symbolic music information retrieval. *3rd International Symposium on Music Information Retrieval (ISMIR)*, Paris, France, p. 31–38, 2002.

ULTSCH, A. Maps for visualization of highdimensional data spaces. In: *In proc. WSOM'03*. Kyushu, Japan: [s.n.], 2003. p. 225–230.



- ULTSCH, A. Emergence in self organizing feature maps. In: *Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)*. [S.l.: s.n.], 2007.
- ULTSCH, A.; HERMANN, L. Architecture of emergent self-organizing maps to reduce projection errors. In: *In Proc. ESANN 2005*. [S.l.: s.n.], 2005. p. 225–230.
- ULTSCH, A.; MOERCHEN, F. Esom-maps: Tools for clustering, visualization, and classification with emergent som. *Technical Report Dept. of Mathematics and Computer Science*, University of Marburg, Germany, n. 46, 2005.
- VATOLKIN, I.; THEIMER, W.; BOTTECK, M. Amuse (advanced music explorer) – a multitool framework for music data analysis. In: *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*. Utrecht, Netherlands: [s.n.], 2010. p. 33–38.
- VIEIRA, M. R.; CHINO, C. T.-J. F.; TRAINA, A. J. M. Dbm-tree: A metric access method sensitive to local density data. In: ANAIS ELETRÔNICOS. *19 o Simpósio Brasileiro de Banco de Dados (SBBD)*. Brasília, DF, 2004. p. 163–177.
- WANG, L. et al. An effective and efficient method for query by humming system based on multi-similarity measurement fusion. In: *2008 International Conference on Audio, Language and Image Processing*. [S.l.: s.n.], 2008. p. 471–475.
- WANGENHEIM, D. rer. nat. Christiane Gresse von. *Critérios e Recomendações Ergonômicas - Aula 3*. Florianópolis: [s.n.], 2017.
- WAZLAWICK, R. S. *Engenharia de Software para Sistemas de Informação: Conceitos e práticas que fazem sentido*. Florianópolis: UFSC, 2012.
- WIKILIVROS. *Processamento de Dados Massivos/Projeto e implementação de aplicações Big Data/Agrupamento baseado em densidade*. 2018? <[https://pt.wikibooks.org/wiki/Processamento\\_de\\_Dados\\_Massivos/Projeto\\_e\\_implementacao\\_](https://pt.wikibooks.org/wiki/Processamento_de_Dados_Massivos/Projeto_e_implementacao_)
- YIANILOS, P. N. Data structures and algorithms for nearest neighbor search in general metric spaces. In: ANAIS ELETRÔNICOS. *4th ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Austin, USA, 1993. p. 311–321.
- YU, H.-M.; TSAI, W.-H.; WANG, H. min. A music retrieval system based on query-by-singing for karaoke jukebox. v. 4182, p. 445–459, 10 2006.
- ZUBEN, P. *Música e tecnologia: o som e seus novos instrumentos*. second. [S.l.]: Irmãos Vitale, 2004.

## APÊNDICE A – Artigo

# Análise de Soluções para Busca por Similaridade (*Matching*) de Dados Musicais

Gisele Bernardes da Silva<sup>1</sup>

<sup>1</sup>Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)  
Caixa Postal 476 – 88.045-108 – Florianópolis – SC – Brazil

`giselebernardes@grad.ufsc.br`

**Resumo.** *Este artigo visa apresentar e comparar soluções para recuperação de informação musical. A intenção é analisar soluções que não necessariamente buscam dados musicais apenas através do casamento direto de parâmetros de entrada para a busca, como título da música, palavras-chave ou um áudio como parte da música, mas também através do casamento aproximado (ou similar) destes parâmetros.*

**Abstract.** *This article aims to present and compare solutions for music information retrieval. The intention is to analyze solutions that do not necessarily search for musical data only through direct marriage of input parameters to the search, such as song title, keywords or an audio with part of the song, but also through approximate (or similar) these parameters.*

## 1. Introdução

A música era um meio de comunicação exclusivamente presencial. No decorrer do tempo, as técnicas e invenções aplicadas ao processo de gravação do som foram surgindo e se aperfeiçoando. Com a Internet, a música ultrapassa os limites físicos da mídia, mergulhando no universo digital. Desta forma, a organização da informação, que inclui a sua representação, tem a principal finalidade de possibilitar a recuperação dessa informação, além da sua guarda para a posteridade. A música se tornou um objeto de consumo universal e extremamente acessível [Daquino 2012, Gomes 2015].

A busca por similaridade musical está inserida dentro de um tema de estudos denominado *Music Information Retrieval*. Os pesquisadores de MIR observam que a motivação maior para essa área de pesquisa é o grande volume de música digital disponível na Internet que, quanto mais cresce, menos possibilita sua recuperação eficiente, visto que estão disponíveis em grande volume, mas sem o tratamento adequado [McLane 1996, Downie 2003, Santini and de Souza 2007].

Este trabalho tem como objetivo geral estudar o estado da arte sobre a recuperação da informação de dados musicais e realizar uma análise comparativa de algumas soluções existentes que não necessariamente buscam dados musicais apenas através do casamento direto de parâmetros de entrada para a busca, mas também através da similaridade destes parâmetros.

Para atingir este objetivo, foi necessário entender os conceitos básicos de som e como o som é transformado em áudio digital ([Muller 2007, Zuben 2004, Pacheco and Lopes 2014]); identificar os formatos de dados musicais e como é feito

o armazenamento deles em bancos de dados ou repositórios digitais ([Semidão 2013, Michels 1992]); e estudar os métodos e algoritmos utilizados para busca por similaridade de dados musicais.

Portanto, a relevância deste trabalho pretende contribuir diretamente, para agregar conhecimento, com o estudo sobre a recuperação da informação de dados musicais, que auxiliarão no desenvolvimento futuro de soluções para busca por similaridade de dados musicais.

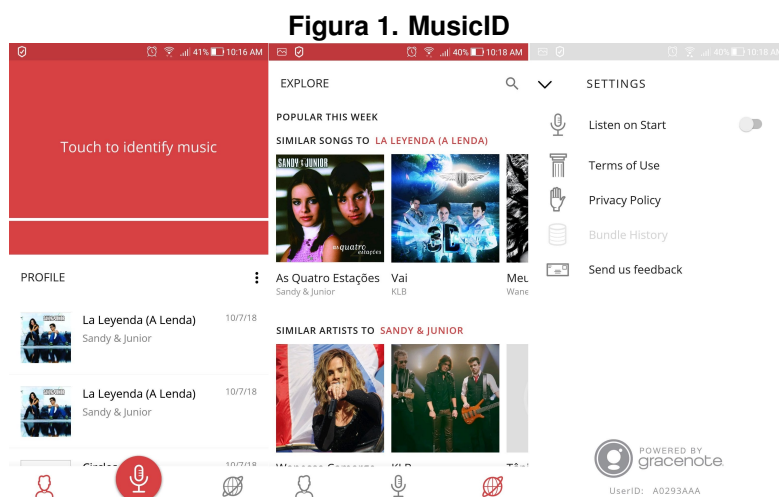
## 2. Soluções Existentes

Nos últimos anos, várias plataformas digitais de *streaming* têm surgido, derivado das intensas procuras por música on-line pelos usuários. Este capítulo apresenta de forma resumida as principais soluções comerciais e da academia, para busca de dados musicais.

### 2.1. Soluções Comerciais

#### 2.1.1. MusicID

Gracenote Inc., fundada em 1998, é uma empresa que fornece metadados de música, vídeo, esportes e tecnologias de reconhecimento automático de conteúdo para empresas e serviços de entretenimento em todo o mundo. A solução da empresa destinada à busca de dados musicais se chama *Gracenote MusicID®* disponível para *smartphones* (ver Figura 1).

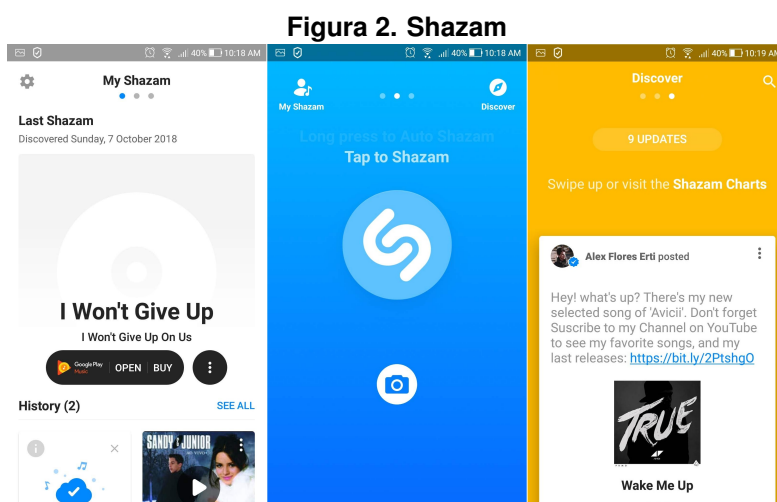


Fonte: Elaborado pela autora

O Gracenote MusicID®, faz o reconhecimento de músicas que são tocadas ao seu redor, combinado ao uso de *fingerprints* e correspondência de texto para identificar arquivos de música digital em um banco de dados mundial de informações musicais. Uma vez reconhecidos, os arquivos são organizados por nome de faixa, nome do álbum e caminhos de pastas e, então, apresentados ao usuário. Ele é um aplicativo para *smartphones* que identifica músicas ouvindo você cantar, cantarolar ou de músicas que são tocadas ao seu redor.

### 2.1.2. Shazam

*Shazam Entertainment Ltd.* foi fundada em 2000 com a idéia de prover um serviço que pudesse conectar as pessoas à música, permitindo a identificação da música através de *smartphones*. A aplicação (ver Figura 2) usa o microfone do *smartphone* ou do computador para capturar uma pequena amostra de música e, então, realiza a identificação da música em um grande banco de dados com mais de 12 bilhões de músicas, com uma alta taxa de acertos.



Fonte: Elaborado pela autora

Para o trecho de música capturado pela aplicação é criado uma *fingerprint*, que é comparada com todas as outras *fingerprints* derivadas das músicas no banco de dados. Se houver uma correspondência, são enviadas informações da música para o usuário, como artista, álbum e título da música.

### 2.1.3. SoundHound

*SoundHound Inc.*, fundada em 2005, é uma empresa pioneira em desenvolvimento de aplicações para reconhecimento de voz, compreensão da linguagem natural, reconhecimento de som e tecnologias de busca.

A plataforma independente de Inteligência Artificial *Houndify*, combinada ao *Automatic Speech Recognition* (ASR) e o *Natural Language Understanding* (NLU), permite ao *SoundHound* a identificação de músicas de forma rápida e eficiente. Seus dois produtos conhecidos no meio musical são:

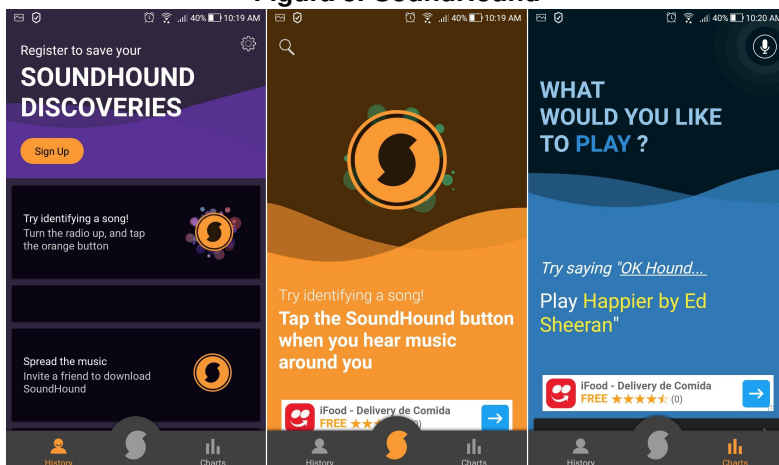
1. *SoundHound Music Search & Play*<sup>1</sup>: aplicativo para *smartphones* onde é possível descobrir, pesquisar e reproduzir qualquer música com controle de voz (ver Figura 3). Ele é um aplicativo para *smartphones* que identifica músicas ouvindo você cantar, cantarolar ou de músicas que são tocadas ao seu redor.

---

<sup>1</sup><https://soundhound.com/soundhound>

2. *Midomi*<sup>2</sup>: aplicação com as mesmas características do item anterior, porém possui versão para *web*. Sua versão *mobile* é destinada a modelos mais antigos de *smartphones*.

Figura 3. SoundHound

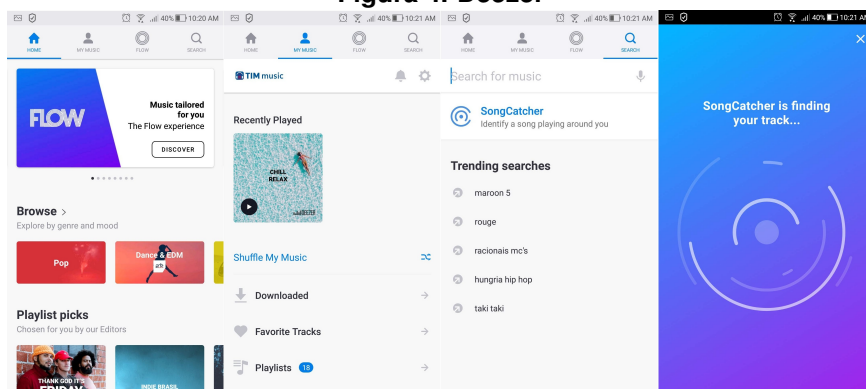


Fonte: Elaborado pela autora

#### 2.1.4. Deezer

Esta solução nasceu da necessidade de facilitar a vida de seu fundador para ouvir e realizar o *download* de músicas. Com isso, o idealizador da plataforma desenvolveu o *Blogmusik.net* em 2006. Devido a sua popularidade, houve objeção de detentores de direitos autores, o que gerou o fechamento do site. Pouco tempo depois, um acordo foi assinado e o antigo site voltou ao ar com o nome de *Deezer* (ver Figura 4).

Figura 4. Deezer



Fonte: Elaborado pela autora

A Deezer (ver Figura 4), também conta com uma série de aplicativos que complementam a experiência musical do usuário. O *Stateeztics*, por exemplo, é um *in-app* exclusivo que traça o perfil musical do usuário e mostra suas estatísticas de consumo a

<sup>2</sup><https://www.midomi.com/>

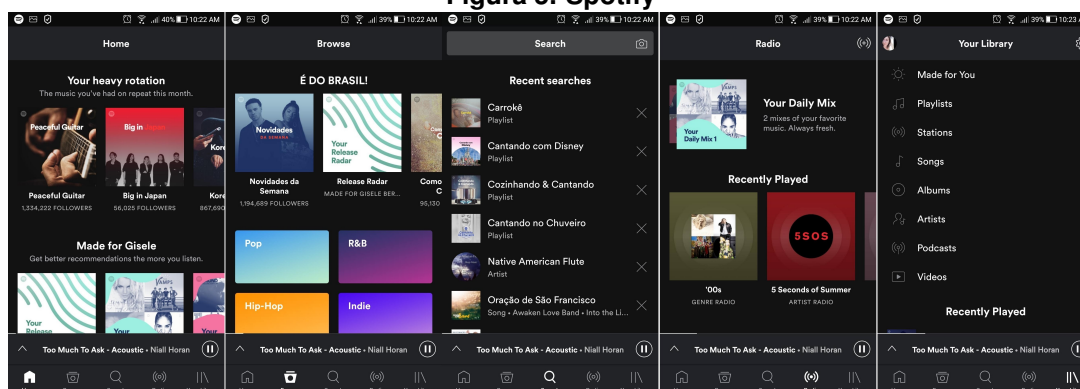
partir do seu histórico sonoro. Outra aplicação disponível é o *Edjing*, que oferece mixagem de músicas com diversas ferramentas de efeitos digitais, além de contar com uma interface bastante intuitiva. Já o usuário que está aprendendo a tocar instrumentos musicais pode contar com o *Chordify*, que reconhece o som que está tocando na Deezer e faz a transcrição automática da harmonia em cifras.

Recentemente, no final do ano de 2017, além da correspondência de texto para identificar arquivos de música digital, a Deezer lançou o seu próprio recurso de identificação de músicas que são tocadas ao seu redor combinado ao uso de *fingerprints*, o *SongCatcher*, desenvolvido pela *ACRCloud* (ver subseção 2.1.8).

### 2.1.5. Spotify

*Spotify Ltd.*, fundada em 2006, é um serviço de *streaming* de música, *podcast* e vídeo, além de ser o mais usado no mundo. A plataforma fornece conteúdo protegido provido de restrição pela gestão de direitos digitais de gravadoras e empresas de mídia (ver Figura 5). O Spotify é um serviço *freemium*: ele possui recursos gratuitos com propagandas ou limitações, e recursos adicionais, como qualidade de transmissão aprimorada e *downloads* de música, que são oferecidos para assinaturas pagas.

Figura 5. Spotify



Fonte: Elaborado pela autora

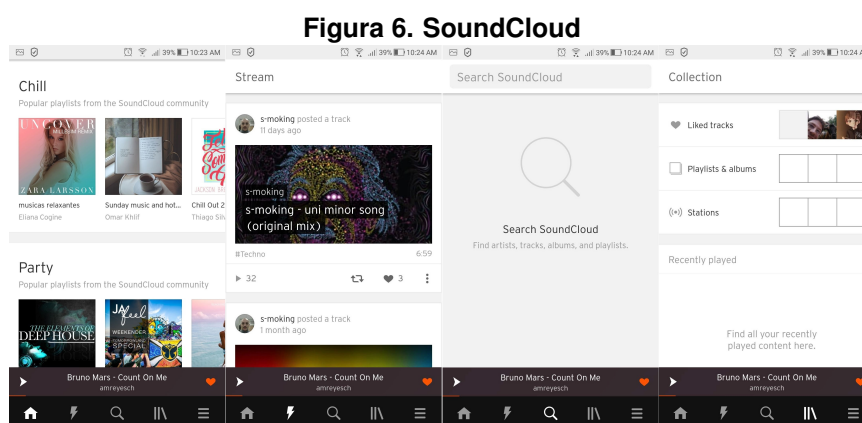
A plataforma emprega um modelo de distribuição de dados híbrido com uma combinação de compartilhamento de dados peer-to-peer<sup>3</sup> (P2P) e uma infraestrutura de servidor. Ao pesquisar uma música através do smartphone e desejar ouvi-lá, o sistema primeiro verifica se a música já se encontra baixada na memória *cache* do *smartphone* para agilizar o processo. Em caso negativo, é feita a conexão diretamente com o servidor do Spotify ao mesmo tempo que o método busca "peers" entre milhões de usuários para que a música que se queira ouvir seja baixada o mais rápido possível. A busca da música é feita pela correspondência de texto para identificar arquivos de música digital através da técnica de recuperação por conteúdo.

<sup>3</sup>Do inglês par-a-par ou simplesmente ponto-a-ponto, é uma arquitetura de redes de computadores onde cada um dos pontos ou nós da rede funciona tanto como cliente quanto como servidor, permitindo compartilhamentos de serviços e dados sem a necessidade de um servidor central.

O Spotify disponibiliza uma *Web API*<sup>4</sup> que permite que desenvolvedores integrem o conteúdo do Spotify em seus próprios aplicativos. O Spotify *Web API* é um serviço com base na arquitetura REST, que retorna em formato JSON dados sobre álbuns, artistas, faixas, playlists, entre outros. Para acessar outras informações é necessária uma autenticação *OAuth*.

### 2.1.6. SoundCloud

*SoundCloud*, criada em 2007, é uma plataforma on-line de publicação de áudio utilizada por profissionais de música (ver Figura 6). Nela os músicos podem colaborar, compartilhar, promover e distribuir suas composições. Originalmente, seu objetivo era permitir que profissionais da música trocassem ideias sobre as composições nas quais estão trabalhando, permitindo uma fácil colaboração e comunicação antes de um lançamento público. Hoje, o site também é utilizado por ouvintes e usuários da *web* em geral.



Fonte: Elaborado pela autora

Os usuários registrados podem ouvir o máximo de conteúdo como quiserem e podem fazer o *upload* de até 180 minutos de áudio ao seu perfil. Todos esses recursos são gratuitos e estão disponíveis para todos os usuários registrados. A plataforma possui uma API integrada a várias aplicações, que permitem fazer o *upload* ou *download* de música e arquivos de música.

O SoundCloud descreve as faixas de música graficamente como formas de onda e permite aos usuários comentar partes específicas do áudio (conhecido como comentários cronometrados). Estes comentários são exibidos ao escutar a parte do áudio que estão se referindo. Outras características incluem respostas, listas de reprodução, seguidores e *downloads* digitais de cortesia. A busca da música é feita pela correspondência de texto para identificar arquivos de música digital através da técnica de recuperação por conteúdo. Não foi encontrada documentação informando como funciona detahadamente a recuperação da música.

---

<sup>4</sup>Do termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos". É uma forma de integrar sistemas, possibilitando benefícios como a facilidade no intercâmbio entre informações com diferentes linguagens de programação.

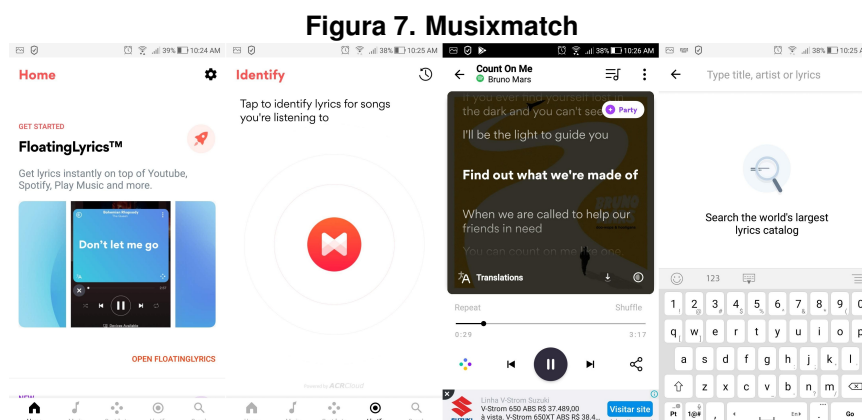


O SoundCloud também disponibiliza uma *Web API* que permite que desenvolvedores integrem o conteúdo do SoundCloud em seus próprios aplicativos. O SoundCloud *Web API* é um serviço com base na arquitetura HTTP, que retorna em formato JSON dados sobre álbuns, artistas, faixas, playlists, entre outros. Para acessar outras informações é necessária uma autenticação *OAuth*.

### 2.1.7. Musixmatch

A *Musixmatch* foi criada em 2010 com o objetivo de mudar a forma como as pessoas experimentam música e letras.

A plataforma pode ser acessada através do site ou via aplicativo para *smartphones*. O Musixmatch digitaliza todas as músicas da biblioteca de música do usuário e encontra letras para todas elas, identificando a letra da música e mantendo sincronizada enquanto a música é tocada. Além da correspondência de texto para identificar arquivos de música digital através da técnica de recuperação por conteúdo, ela possui também a capacidade para capturar uma pequena amostra de música através de *fingerprints* (mesma função encontrada em soluções como o *Shazam*), desenvolvido pela ACRCLOUD.



Fonte: Elaborado pela autora

### 2.1.8. ACRCLOUD

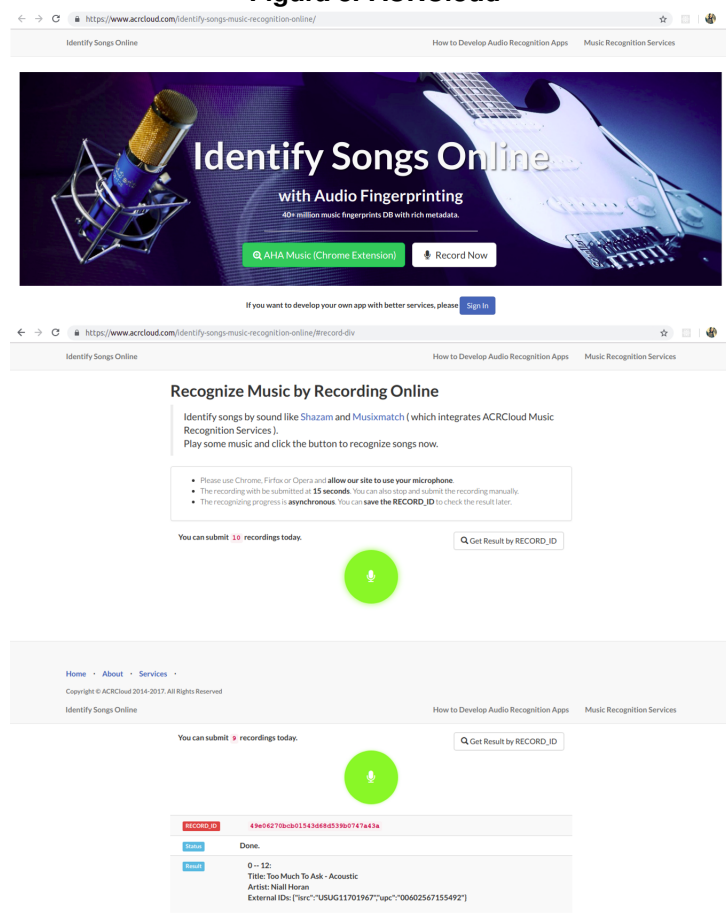
A *ACRCLOUD* foi criada em 2015, sendo a solução vitoriosa no campeonato de *Audio Fingerprinting* do MIREX2015, organizado pelo Laboratório Internacional de Avaliação de Sistemas de Recuperação de Informação Musical (IMIRSEL, sigla em inglês).

ACR (*Automatic Content Recognition*) é uma tecnologia de identificação para reconhecimento de conteúdo reproduzido em um dispositivo de mídia. Ele permite que usuários obtenham rapidamente informações detalhadas sobre o conteúdo que acabaram de experimentar sem qualquer entrada de texto ou esforço de pesquisa.

ACRCLOUD (ver Figura 8) é uma plataforma de microserviços na nuvem que possui reconhecimento de música através de *fingerprints*, onde identifica músicas ouvindo você cantar ou de músicas que são tocadas ao seu redor, além do monitoramento de transmissão

com identificação e apresentação de conteúdo, entre outros. Ele possui integração com serviços de música como o Spotify, Deezer, entre outros, que permite desenvolvedores acessarem diretamente esses serviços e oferecer links diretos para seus usuários.

**Figura 8. ACRCLOUD**



Fonte: [ACRCLOUD 2015], elaborado pela autora

### 2.1.9. Musipedia

Musipedia é uma enciclopédia aberta de música, criação inspirada no Wikipedia<sup>5</sup>, para localização, edição e expansão de coleções de tons, melodias e temas musicais (ver Figura 9). A enciclopédia utiliza o mecanismo de pesquisa de melodias, do qual chamam de *melodyhound*, onde é possível encontrar e identificar uma música, mesmo que a melodia seja tudo o que você saiba no momento. A busca também pode ser feita através da pesquisa de contorno melódico (Código de Pearson) ou com base no ritmo. Ainda, os conteúdos podem ser alterados por qualquer usuário, podendo conter um pedaço de música, um arquivo MIDI, informações textuais sobre o trabalho e o compositor.

É possível também integrar a pesquisa do Musipedia ao seu próprio serviço web, utilizando as interfaces SOAP, que possibilitam pesquisar com base na melodia, no contorno melódico ou no ritmo.

<sup>5</sup><https://www.wikipedia.org/>

Figura 9. Musipedia



Fonte: [Development 2018], elaborado pela autora

## 2.2. Soluções Acadêmicas

### 2.2.1. AMUSE

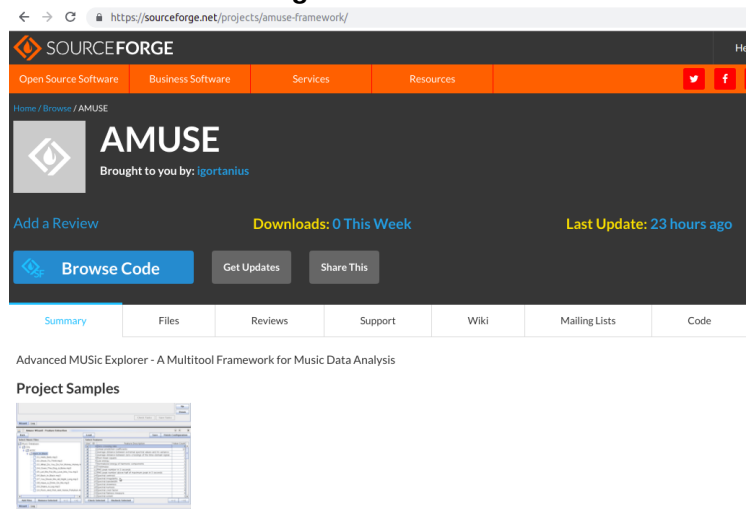
AMUSE (*Advanced Music Explorer*)<sup>6</sup> é um *framework* desenvolvido pela TU Dortmund, na Alemanha, licenciado sob a GPL e implementado em JAVA. Portanto, ele pode ser executado em qualquer sistema operacional que suporte o Java Runtime Environment (ver Figura 10).

Segundo [Vatolkin et al. 2010], o *framework* fornece diferentes funcionalidades, como:

- Processamento de som, convertendo arquivos de áudio MP3 em ondas sonoras;
- *Downsampling* e estéreo para a conversão de arquivos de áudio mono;
- Divisão automática de arquivos wave;
- Escalabilidade usando multi-threading em uma máquina ou fornecendo as tarefas para sistemas de grade como Sun Grid Engine ou LSF Batch;
- Gerenciamento eficiente do conjunto de dados que suporta diretamente o formato WEKA ARFF;
- Componente logger integrado.

<sup>6</sup><https://sourceforge.net/projects/amuse-framework/>

**Figura 10. AMUSE**



Fonte: [Dortmund 2018], elaborado pela autora

O AMUSE possui subtarefas em cadeia para recuperação da informação musical. Cada tarefa pode ser calculada em várias unidades de processamento. Inicia-se pela tarefa de extração de recursos, que fornece descritores numéricos de baixo nível ou alto nível do sinal de áudio (por exemplo, extração de melodia da música). Depois que a tarefa de extração é carregada na memória, é realizado o processamento dos recursos, em uma etapa intermediária, que serve de entrada para a técnica de classificação. Por fim, é realizada a validação dos resultados da classificação.

As ferramentas integradas não têm restrições de uso em relação aos seus códigos-fonte. Se eles não estiverem disponíveis como bibliotecas Java, as versões executáveis deverão ser fornecidas. Nesse caso, pode certamente levar à dependência do sistema operacional em execução.

O projeto é oferecido gratuitamente à comunidade de pesquisa. Informações mais detalhadas sobre o projeto podem ser consultadas em [Vatolkin et al. 2010] e [Dortmund 2018].

### 2.2.2. CLAM

CLAM (*C++ Library for Audio and Music*)<sup>7</sup> é um *framework* desenvolvido em C++ no *Music Technology Group* (MTG) da Universidade Pompeu Fabra em Barcelona, Espanha. Ele oferece uma plataforma completa de desenvolvimento e pesquisa para o domínio de áudio e música baseado na técnica SMS. Além de oferecer um modelo abstrato para sistemas de áudio, ele também inclui um repositório de algoritmos de processamento e tipos de dados, bem como diversas ferramentas, como entrada/saída de áudio ou MIDI (ver Figura 11).

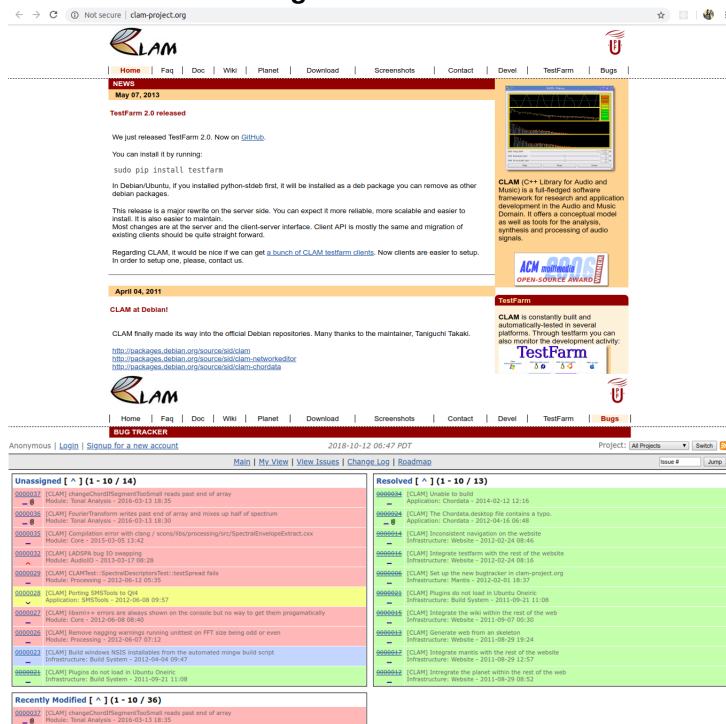
Segundo [Amatriain 2004], as características mais importantes do framework são:

- Verdadeiramente orientado a objetos. Extensas técnicas de engenharia de software

---

<sup>7</sup><http://clam-project.org/>

Figura 11. CLAM



Fonte: [Development 2004], elaborado pela autora

foram aplicadas para projetar uma estrutura que seja altamente (re)utilizável e compreensível;

- É abrangente, uma vez que não só inclui classes para processamento, mas também para entrada e saída de áudio e MIDI, serviços de serialização XML, algoritmos e visualização e interação de dados, e manipulação multi-threading;
- Lida com uma ampla variedade de tipos de dados extensíveis que vão desde sinais de baixo nível (como áudio ou espectro) até estruturas semânticas de nível superior (como frase musical ou segmento);
- É multiplataforma. Todo o código é ANSI C ++ e é regularmente compilado no Linux, Windows e Mac OSX usando os compiladores mais usados. Até mesmo o código para entrada/saída, visualização e multithreading é de plataforma cruzada até a camada mais baixa possível;
- O projeto está licenciado sob os termos e condições GPL (Licença Pública GNU). Apesar de possuir a opção de licenciamento duplo da estrutura (ou seja, oferecer uma licença comercial alternativa), tudo o que é oferecido na versão pública é GPL e o projeto é, portanto, Software Livre, código aberto e colaborativo;
- Base para todos os desenvolvimentos futuros no MTG e, portanto, mantido e atualizado regularmente;
- O framework pode ser usado como uma biblioteca C ++ regular ou como uma ferramenta de prototipagem. No primeiro modo, o usuário pode estender, adaptar ou otimizar a funcionalidade da estrutura para implementar um aplicativo específico. No segundo modo, o usuário pode facilmente construir um protótipo para testar um novo algoritmo ou aplicativo de processamento de sinais.

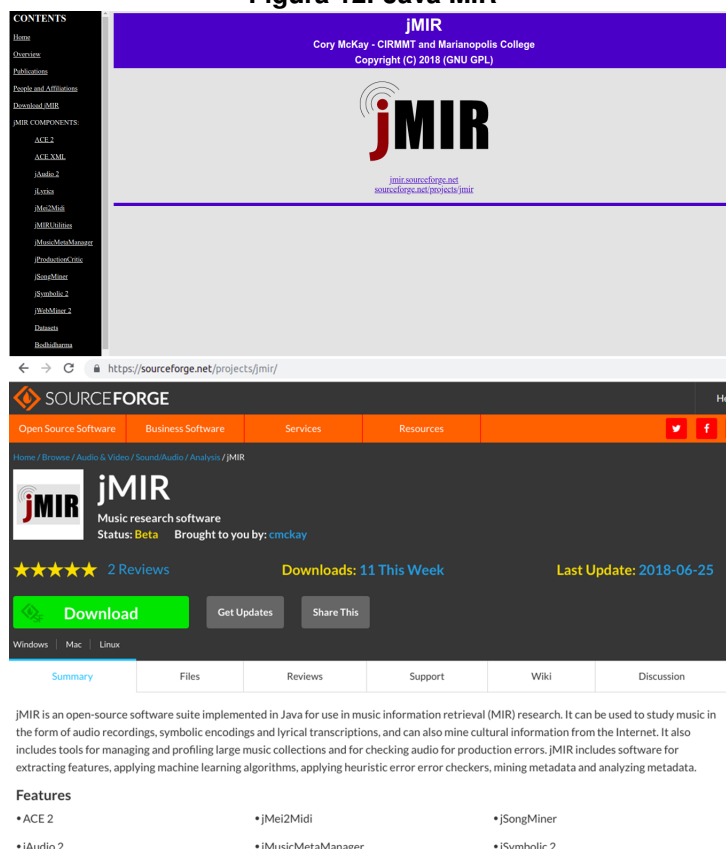
Informações mais detalhadas sobre este *framework* podem ser consultadas em

[Amatriain 2007, Amatriain 2004].

### 2.2.3. Java MIR

jMIR (Java MIR)<sup>8</sup> é um *software* que possui um conjunto de componentes desenvolvido na CIRMMT e Marianopolis College, ambos localizados no Canadá. Cada um dos componentes pode ser utilizado separadamente ou como um todo (ver Figura 12).

### Figura 12. Java MIR



Fonte: [McKay 2018], elaborado pela autora

O software é de código livre implementado em Java para uso nas pesquisas de Recuperação de Informação Musical (MIR) e se baseia em uma técnica de mineração de dados, a classificação. Ele pode ser usado para estudar música na forma de gravações de áudio, codificações simbólicas e transcrições líricas, e também pode extrair informações culturais da Internet. Ainda, ele inclui ferramentas para gerenciar e criar perfis de grandes coleções de músicas e para verificar o áudio quanto a erros de produção. É bem documentado e inclui GUIs para aumentar a usabilidade geral.

O objetivo principal do *software* é auxiliar nas pesquisas em classificação automática de música e a análise de similaridade, proporcionando as seguintes características:

<sup>8</sup><http://jmir.sourceforge.net/>

- Tornar tecnologias sofisticadas de reconhecimento de padrões acessíveis a pesquisadores de música com históricos técnicos e não técnicos;
- Eliminar duplicação redundante de esforço;
- Aumentar a cooperação e a comunicação entre os grupos de pesquisa;
  - Facilitar o desenvolvimento iterativo e o compartilhamento de novas tecnologias MIR;
  - Facilitar comparações objetivas de algoritmos.
- Facilitar a pesquisa combinando características musicais de alto nível, baixo nível e culturais (ou seja, características simbólicas, áudio e web-minadas).

Informações mais detalhadas sobre o projeto estão disponíveis nas publicações acadêmicas<sup>9</sup>. Manuais e documentação para cada componente também podem ser consultados em [McKay 2018] e [McKay 2010].

#### 2.2.4. MIRtoolbox

*MIRtoolbox*<sup>10</sup> é um pacote de ferramentas escritas em Matlab para a extração de recursos musicais, como tonalidade e ritmo, tanto para especialistas quanto para não especialistas do Matlab (ver Figura 13). Ele foi desenvolvido dentro do contexto de um Projeto Europeu chamado “Tuning the Brain for Music”, dedicado ao estudo da música e da emoção, com colaboração entre neurociências, psicologia cognitiva e ciência da computação. Os grupos e instituições envolvidos são a Music Cognition Team da University of Jyväskylä na Finlândia e o Music Acoustics Group do KTH em Estocolmo.

Segundo [Lartillot 2013], foi elaborado um manual onde são descritas as seguintes especificações da solução:

- Quadro modular: É baseado em um conjunto de blocos de construção que podem ser parametrizados, reutilizados, reordenados, etc.;
- Sintaxe simples e adaptativa: Os usuários podem se concentrar no design geral e o MIRtoolbox que cuida das tarefas laboriosas subjacentes;
- Software livre e código-fonte aberto: A ideia é propor a capitalização da expertise da comunidade de pesquisa e oferecê-la de volta à comunidade;
- Recursos: O MIRtoolbox inclui cerca de 50 extratores de recursos de áudio e música e descritores estatísticos.

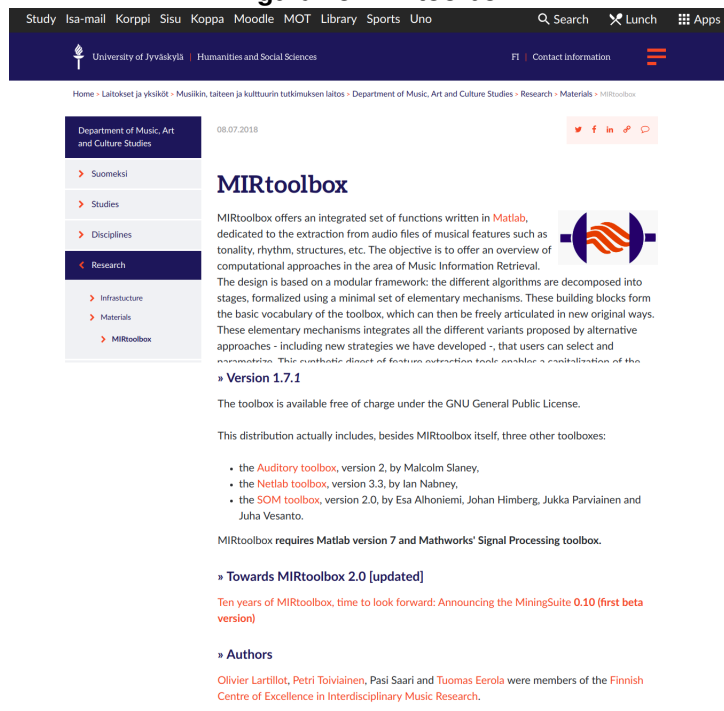
O MIRtoolbox é baseado em técnicas de mineração de dados e, por ser um pacote de ferramentas, são variadas as técnicas para recuperação da informação musical, como a classificação.

Desta forma, ele pode ser útil para a comunidade de pesquisa em Recuperação da Informação Musical (MIR), mas também para fins educacionais. Mais informações sobre o projeto podem ser consultadas em [Lartillot and Toivainen 2007, Lartillot 2013] e [of Jyväskylä 2018]

<sup>9</sup><http://jmir.sourceforge.net/publications.html>

<sup>10</sup><https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox>

Figura 13. MIRtoolbox



Fonte: [of Jyväskylä 2018], elaborado pela autora

### 2.2.5. MusicMiner

O *Databionic MusicMiner*, desenvolvido como parte de um projeto de pesquisa do Grupo de Pesquisa em Databionics da Universidade de Marburg, na Alemanha, é um navegador para dados musicais baseado em técnicas de mineração de dados, como clusterização e visualização com base no paradigma dos mapas geográficos ESOM (ver Figura 14).

A coleção de músicas é recuperada e apresentada em forma de mapa topográfico com pequenos pontos para as músicas. O usuário pode interagir com o mapa de diferentes formas para selecionar e ouvir músicas; explorar suas músicas; e criar playlists baseadas no paradigma de mapas geográficos.

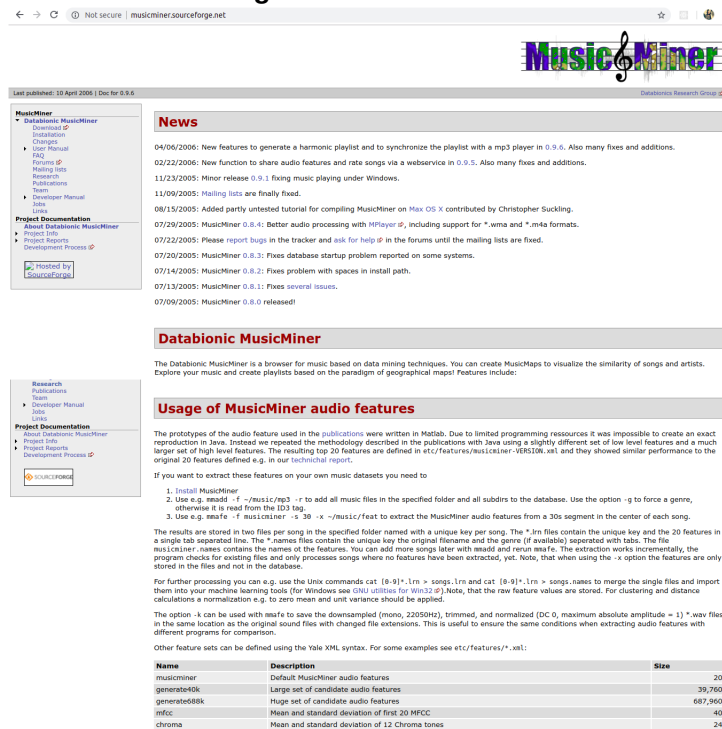
O site do projeto<sup>11</sup> apresenta as seguintes características da solução:

- Análise automática de uma árvore de pastas com arquivos de música (MP3, OGG, WMA, M4A, MP2, WAV);
- Descrição automática de arquivos de áudio digital por som;
- Criação de *MusicMaps* para navegar pelo espaço sonoro com base no paradigma dos mapas geográficos ESOM;
- Criação visual de *playlists*;
- Pesquisa por similaridade na coleção de músicas com base no som;
- Navegação hierárquica personalizável da base de dados, como por exemplo, por gênero/artista/álbum ou ano/artista;
- Base de dados flexível, incluindo o armazenamento separado de vários artistas por música, álbuns e listas de reprodução como parte de uma lista de reprodução;

<sup>11</sup><http://musicminer.sourceforge.net/>



Figura 14. MusicMiner



Fonte: [of Marburg 2005], elaborado pela autora

- Importação e exportação de meta informações baseadas em XML.

O *MusicMiner* é escrito em Java para máxima portabilidade e publicado sob os termos da GPL (General Public License). Seu foco principal é a pesquisa e o ensino. Informações mais detalhadas sobre o projeto podem ser encontradas em [Morchén et al. 2005] e [of Marburg 2005].

## 2.2.6. Tunebot

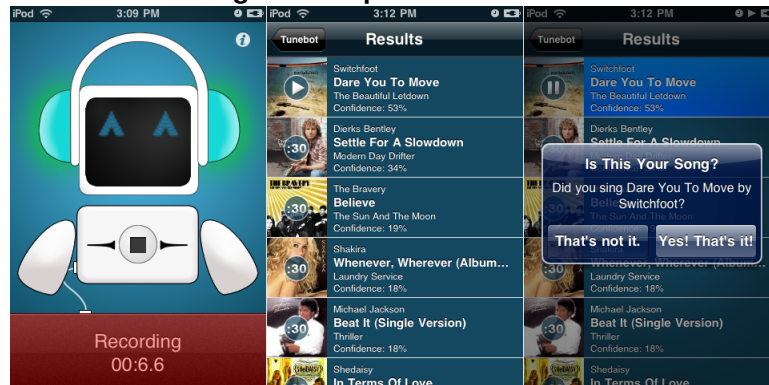
*Tunebot*<sup>12</sup> é um projeto criado em 2015, desenvolvido e mantido por *Interactive Audio Lab*<sup>13</sup> na Universidade de Northwestern, nos Estados Unidos. Segundo [Huq et al. 2010b], o Tunebot está disponível como um serviço web (ver Figura 16) e está atualmente em teste beta como um aplicativo do iPhone (ver Figura 15). A interação do usuário nas versões da Web e do iPhone é idêntica: (i) cante, e (ii) escolha. O usuário simplesmente canta uma parte da música desejada para o Tunebot e o sistema retorna uma lista ordenada de músicas. Cada música pode ser reproduzida por um simples clique. Enquanto a música está tocando, o sistema apresenta uma caixa de diálogo perguntando se esta é a música correta. Se o usuário clicar em "sim", a consulta será armazenada no banco de dados como um exemplo pesquisável para essa música. O usuário é conectado à Amazon.com ou ao iTunes, onde a música pode ser comprada.

O sistema não exige chaves de pesquisa codificadas manualmente, pois atualiza

<sup>12</sup><http://music.cs.northwestern.edu/data/tunebot/>

<sup>13</sup><http://music.eecs.northwestern.edu/>

Figura 15. Aplicativo Tunebot



Fonte: [Lab 2008a], elaborado pela autora

Figura 16. Tunebot Web



Fonte: [Huq et al. 2010b]

automaticamente o banco de dados com novas chaves de pesquisa derivadas de consultas e contribuições do usuário [Huq et al. 2010b]. O banco de dados do *Tunebot* compara as músicas com as músicas cantadas pelos usuários utilizando o algoritmo *query by humming*.

Outro objetivo do projeto é ajudar pesquisadores na área de reconhecimento de músicas que utilizam o algoritmo *query by humming*, facilitando uma pesquisa mais precisa do desempenho do mundo real do que seria possível com conjuntos de dados existentes.

Informações mais detalhadas sobre o projeto podem ser consultadas em

[Huq et al. 2010b, Cartwright and Pardo 2012] e [Huq et al. 2010a].

### 3. Análise Comparativa

Esta seção apresenta os critérios definidos para a análise das soluções existentes, a análise comparativa propriamente dita e os resultados obtidos com a análise. A metodologia de comparação entre as soluções leva em conta documentações públicas disponíveis que possuam informação pertinente.

#### 3.1. Critérios de Análise

Com base nos conceitos apresentados por [Wazlawick 2012], que define os atributos de qualidade internos, externos e de uso de produtos de software; e pela autora [rer. nat. Christiane Gresse von Wangenheim 2017], que apresenta heurísticas definidas por [Nielsen 1995] para assegurar que os produtos são usáveis, esta seção tem por objetivo apresentar os critérios que devem ser considerados na avaliação das soluções para busca de dados musicais, de modo a compará-las e facilitar a escolha pela mais apropriada para uma determinada situação.

Neste trabalho foi realizada uma adaptação de heurísticas para permitir uma boa análise da eficiência e adequação funcional das soluções apresentadas, sendo utilizado informações disponibilizadas nas documentações próprias de cada solução comercial e acadêmica. Já para a análise da usabilidade foi realizada uma adaptação do *MATcH Checklist* disponibilizado pelo Grupo de Qualidade de Software (GQS) da UFSC, onde o conjunto de perguntas possui uma escala de resposta com 2 opções: Sim (a solução atende o objetivo) e Não (a solução não atende o objetivo).

Assim sendo, os seguintes critérios e subcritérios são considerados:

- **Eficiência:** Trata da otimização do uso de recursos de tempo e espaço. Espera-se que o sistema seja o mais eficiente possível de acordo com o tipo de problema que ele soluciona:
  - *Comportamento em relação ao tempo:* Mede o tempo que o sistema leva para processar suas funções, ou seja, o tempo de reconhecimento/busca de uma música;
  - *Utilização de recursos:* Avalia a complexidade das estratégias e algoritmos utilizados na recuperação de informação musical;
  - *Bitrate:* Mede a qualidade do áudio. Essa qualidade consiste no número médio de bits que será comprimido em um segundo de dados. A unidade utilizada é o KBPS ou 1000 BITS por segundo;
- **Adequação Funcional:** Mede o grau no qual o produto oferece funções que satisfazem necessidades estabelecidas e implicadas quando o produto é usado sob condições especificadas:
  - *Disponibilidade:* Avalia a disponibilidade da aplicação em diferentes plataformas;
  - *Modelo de desenvolvimento:* Avalia se a solução é de código aberto, dando a possibilidade para que qualquer um consulte, examine ou modifique o produto;
  - *Integrações:* Avalia se a solução permite extensões e/ou integrações com outras aplicações;

- *Acessibilidade*: Avalia se a solução possui acesso ao acervo de músicas on-line e/ou off-line;
- *Busca de dados*: Avalia se a solução foi projetada para *matching* exato ou por similaridade (aproximado);
- *Inclusão da dados*: Avalia se a solução permite o envio de músicas feito pelo usuário;
- *Modelo de pagamento*: Avalia o custo da solução, como por exemplo: Gratuito, Pago ou Freemium.
- Usabilidade: Avalia o grau no qual o produto tem atributos que permitem que seja entendido e que seja atraente ao usuário, quando usado sob condições especificadas:
  - *Visibilidade do status do sistema*: O sistema deve sempre manter o usuário informado sobre o que está acontecendo. Por exemplo, os componentes interativos selecionados são claramente distintos dos demais?
  - *Prevenção de erros*: Mensagens de erros devem ser claras e objetivas, devem indicar o problema com precisão e sugerir uma solução;
  - *Flexibilidade e eficiência de uso*: A solução deve permitir configuração de ações frequentes. Por exemplo, as funções mais utilizadas são facilmente acessadas?
  - *Estética e Design minimalista*: Mensagens de diálogos não devem conter informações irrelevantes. Por exemplo, o menu é esteticamente simples e claro, com opções fáceis de encontrar, dispostas em uma ordem lógica e com títulos curtos?
  - *Pouca interação homem/dispositivo*: Qualquer informação deve ser fácil de pesquisar e deve ser focada na tarefa do usuário. Por exemplo, a navegação da solução é intuitiva, é fácil chegar à tela desejada?

## 3.2. Análise

Uma comparação é primeiramente realizada entre as soluções comerciais, seguida das soluções acadêmicas.

### 3.2.1. Soluções Comerciais

Inicialmente, para a análise das soluções comerciais, foram criadas 3 tabelas, uma para cada grande critério analisado para melhor visualização.

Em relação ao critério de Eficiência mostrado na Tabela 1, o comportamento das soluções em relação ao tempo foram constatados as diferenças verificando o tipo de método utilizado para o reconhecimento/busca de músicas. Os testes foram realizados usando conexão de Internet Wi-Fi e Dados Móveis (4G). Com um smartphone e um cronômetro, foram feitas 5 execuções em cada aplicação com músicas diferentes. E então, calculado a média de todos os tempos para cada aplicação.

Para as soluções utilizando RPC, foi testada a busca de músicas pela correspondência de texto através de metadados (ou seja, através da informação do título, álbum ou gênero da música). Neste caso, a busca demorou até 30s para retornar uma amostra de resultados. Para as soluções que utilizam *Fingerprint* e fazem reconhecimento de música

de forma exata (E), foi testada a busca através da informação de uma parte de áudio "original". Já para as soluções de reconhecimento de músicas de forma aproximada (A), foi testada a busca através da informação de uma parte de áudio "original", do canto ou do cantarolar. Ambas as soluções podem variar de 5s até 13s para retornar uma amostra de resultados.

Cabe observar aqui que uma análise feita apenas sobre o comportamento geral das soluções em relação ao tempo não seria relevante, já que tempos entre 5s e 30s são relativos à situação em que o usuário está no momento. Por exemplo, o usuário pode estar em lugares onde a conexão de Internet não é boa ou quase nula, ou pode estar em lugares onde a conexão de Internet é ótima, já que o reconhecimento de uma música depende de uma conexão de Internet. Quanto melhor a conexão, mais rápido será o reconhecimento da música.

É possível verificar que tanto soluções para o reconhecimento de músicas de forma aproximada, como o MusicID e o ACRCLOUD, quanto soluções para o reconhecimento de forma exata, como o Shazam, Deezer e Musixmatch, podem utilizar o mesmo método de *Fingerprint*. Apesar de utilizarem o mesmo método, a forma como pode ter sido desenvolvido é que gerou a vantagem competitiva dentre os concorrentes do mesmo ramo.

O SoundHound utiliza a tecnologia de Inteligência Artificial (IA) para o reconhecimento de músicas. O desenvolvimento de uma solução baseada em IA se dá através de métodos e algoritmos de aprendizagem de máquina, como Classificação e Clustering, porém não foi encontrada documentação que especificasse o método ou algoritmo utilizado pelo SoundHound para o reconhecimento das músicas. Com essa solução, é possível recuperar músicas através da voz, cantarolando ou até mesmo informando uma parte do áudio "original". Conforme os usuários cantam e conforme a escolha da música do resultado amostrado, a IA associa a cantoria àquela música. Então, quanto mais cantar e buscar as músicas "certas" para a cantoria, a taxa de acerto aumenta, formando uma rede totalmente interligada para o reconhecimento correto de músicas.

Há também as soluções que utilizam o método de Recuperação por Conteúdo (RPC), que seriam as cadeias de caracteres, em buscas, por exemplo, por título, álbum ou gênero da música - o que chamamos de metadados -, conforme já explicitado em seções anteriores deste trabalho. Destas soluções, pode-se citar o Spotify e o SoundCloud. A busca das músicas no Spotify, por exemplo, é feita exclusivamente através de texto, não sendo possível adicionar músicas criadas pelo usuário. Entretanto, é possível criar playlists com as suas músicas preferidas.

Para o Musipedia, não foi encontrada documentação que especificasse o método ou algoritmo utilizado para o reconhecimento das músicas.

Quanto ao critério de Bitrate, das 9 soluções analisadas, foi encontrado essa informação em apenas 4. Diferentes bitrates podem produzir qualidade de som variável. Assim, quanto maior o bitrate, mais vezes por segundo o som original é amostrado, produzindo assim uma reprodução mais autêntica e melhor do som [SanDisk 2018].

O Deezer e Spotify apresentam bitrate de até 320kbps, possuindo 3MB por minuto de áudio e uma qualidade aproximada à de um CD. Já o MusicID e o SoundCloud apresentam bitrate de até 128kbps, possuindo 1MB por minuto de áudio e uma qualidade mínima aceitável em termos de alta fidelidade do áudio [F 2012].

**Tabela 1. Análise das soluções comerciais: Critério de Eficiência**

CRITÉRIOS		Eficiência		
		Comportamento em relação ao tempo	Utilização de recursos	Bitrate
C	<b>MusicID</b>	Até 8s	FP	Até 128kbps
O	<b>Shazam</b>	Até 8s	FP	-
M	<b>SoundHound</b>	Até 5s	IA	-
E	<b>Deezer</b>	Até 10s	RPC e FP	Até 320kbps
R	<b>Spotify</b>	Até 30s	RPC	Até 320kbps
C	<b>SoundCloud</b>	Até 30s	RPC	Até 128kbps
I	<b>Musixmatch</b>	Até 13s	RPC e FP	-
A	<b>ACRCloud</b>	Até 5s	FP	-
L	<b>Musipedia</b>	-	-	-

**Tabela 2. \***

Legenda: FP - Fingerprint; IA - Inteligência Artificial; RPC - Recuperação por Conteúdo;

Em relação ao critério de Adequação Funcional mostrado na Tabela 3, das 9 soluções analisadas, 7 são multiplataformas, sendo em sua maioria voltadas para uso em smartphones e 2 de uso exclusivo via Web.

Pode-se verificar que as soluções comerciais, em sua grande maioria, possuem um modelo de desenvolvimento fechado (F), ou seja, seu código não pode ser alterado, mas disponibilizam APIs Web para a comunidade de desenvolvedores, para que a solução possa ser incorporada a seus próprios sites e aplicações.

Quanto à acessibilidade, todas precisam de conexão com a Internet para o uso de suas funcionalidades e apenas 2 das 9 soluções analisadas possuem acesso off-line, que se dá através do download de músicas para a memória interna do dispositivo. As 2 soluções são Spotify e Deezer.

Das 9 soluções comerciais analisadas, 4 possuem reconhecimento de músicas de forma aproximada (A), ou seja, é possível realizar buscas de músicas por similaridade através da voz, sendo que 2 delas permite cantarolar. Já as 5 demais soluções realizam o reconhecimento de músicas de forma exata (E), sendo necessária uma parte de áudio "original" e/ou uma busca por conteúdo através de metadados.

Apenas 3 soluções permitem a inclusão de músicas criadas por usuários, enquanto as outras 6 permitem a inclusão de novas músicas somente através de contatos com gravadoras e/ou artistas. Destas 3 primeiras, duas reconhecem músicas de forma aproximada: ACRCloud e Musipedia. A primeira é um serviço na nuvem, sendo, até o momento, o maior banco de dados de músicas e um serviço utilizado pela maioria das outras soluções aqui analisadas, como Musixmatch, que permite o reconhecimento de músicas por similaridade, e o Musipedia é uma wikipedia de músicas que aceita contribuições musicais de diversas formas: através da voz, partes de músicas, ou até em formato MIDI (quando o som é criado digitalmente).

Com exceção do ACRCLOUD, todas as demais soluções possuem versões gratuitas para uso, com a possibilidade de pagar uma mensalidade e não ter interrupções e propagandas entre as músicas. Ainda, poucas soluções possuem integrações com outros serviços e/ou aplicações. Com o Shazam, por exemplo, é possível integrar-se ao Spotify e, então, ao encontrar uma música, poder ouvi-la por completo. Da mesma forma para o SoundHound, além de ser possível o compartilhamento da sua pesquisa com o Twitter. Com o Musixmatch integrado ao Spotify, por exemplo, ao encontrar uma música, você acompanha a música com a letra em tempo real. O ACRCLOUD e o Musixmatch são as soluções que possuem o maior número de integrações.

**Tabela 3. Análise das soluções comerciais: Critério de Adequação Funcional**

CRITÉRIOS		Adequação funcional						
		Disponibilidade	Modelo de desenv.	Integrações	Acessibilidade	Busca de dados	Inclusão de dados	Modelo de Pagamento
C O M E R C I A L	<b>MusicID</b>	iOS, Android	F	não	On	A	não	G
	<b>Shazam</b>	iOS, Android	F	Spotify, Google Music, Apple Music	On	E	não	G
	<b>SoundHound</b>	iOS, Android	F	Spotify, Youtube, Twitter	On	A	não	F
	<b>Deezer</b>	iOS, Android, Windows, Web	F	Google Music, Twitter, Facebook	On/Off	E	não	F
	<b>Spotify</b>	iOS, Android, Windows, Web Linux, OS X	F	não	On/Off	E	não	F
	<b>SoundCloud</b>	iOS, Android	F	não	On	E	sim	F
	<b>Musixmatch</b>	iOS, Android	F	Spotify, Deezer, Google Music, Youtube, Twitter, Facebook, Last.fm	On	E	não	F
	<b>ACRCLOUD</b>	Web	F	Spotify, Deezer, Youtube, ISRC, UPC, LyricFind, Music Story, SyncPower	On	A	sim	P
	<b>Musipedia</b>	Web	A	não	On	A	sim	G

**Tabela 4. \***

Legenda: F - Fechado; A - Aberto; E - Busca Exata; A - Busca Aproximada; G - Gratuito; F - Freemium; P - Premium;

Com relação à usabilidade mostrado na Tabela 5, testes foram realizados considerando os critérios e subcritérios descritos na seção 3.1. Foi realizada uma adaptação do *MATCH Checklist* disponibilizado pelo Grupo de Qualidade de Software (GQS) da UFSC, onde o conjunto de perguntas possui uma escala de resposta com 2 opções: Sim

(a solução atende o objetivo) e Não (a solução não atende o objetivo). Cada solução possui particularidades que podem atrair ou afastar o usuário. Os resultados dos testes são discriminados abaixo, feitos em um smartphone, modelo Asus Zenfone 4, com sistema operacional Android.

Todas as soluções são simples, possuem navegação intuitiva e são de fácil execução. As funções mais utilizadas são facilmente acessadas. Os componentes interativos são claramente distintos uns dos outros, com ícones intuitivos. Com exceção do Musipedia, as soluções possuem uma linguagem clara, concisa e funcionam corretamente, não apresentando problemas. Com relação à Musipedia (ver Figura 9), ela é uma enciclopédia de músicas e sua navegação não é muito intuitiva, além de possuir textos longos, não apresentando uma linguagem clara e concisa. Não funciona corretamente, não apresentando mensagens de erros claras e objetivas.

No caso do SoundHound, Deezer e Spotify, o menu é esteticamente simples, mas suas versões gratuitas possuem propagandas, o que polui a tela. Ao se adquirir a versão paga, as propagandas são retiradas. Já o MusicID e o Musixmatch não possuem muitas funcionalidades e o menu é esteticamente simples, claro e sem propagandas.

Deezer, Spotify e SoundCloud executam buscas de músicas pela correspondência de texto (metadados) para identificar arquivos de música digital, ou seja, através da informação do título, álbum ou gênero da música.

O MusicID (ver Figura 1) é de uso exclusivo para o reconhecimento de músicas ouvindo você cantar, cantarolar ou que são tocadas ao seu redor. O mesmo vale para o SoundHound (ver Figura 3), com o diferencial de possuir mais funcionalidades além do reconhecimento de músicas, como a possibilidade de criar playlists e o compartilhamento com o Twitter. Além das informações básicas sobre a música, o aplicativo sugere alguns vídeos que podem ser assistidos diretamente no SoundHound.

O Shazam (ver Figura 2) é similar ao MusicID e ao SoundHound. A diferença é que faz o reconhecimento de músicas apenas tocadas ao seu redor. Ele também possui a funcionalidade de tirar fotos de QRCodes para realizar a busca da música. O Deezer (ver Figura 4) é similar ao Shazam, pois também possui a funcionalidade de reconhecimento de músicas que são tocadas ao seu redor e a possibilidade do download das músicas para uso off-line, além da criação de playlists.

O Spotify (ver Figura 5) é similar ao Deezer em termos de funcionalidades. A diferença é que o Spotify não possui reconhecimento das músicas pelo som (trecho de áudio ou voz). Para usufruir de todas as possibilidades do aplicativo, como o download de músicas para uso off-line, é necessário obter a versão premium. Por sua vez, o SoundCloud (ver Figura 6) é similar ao Spotify, com a diferença de ser o único aplicativo comercial analisado que permite a inclusão de músicas criadas pelos usuários.

O Musixmatch (ver Figura 7) é um aplicativo que sincroniza letras de músicas e também permite o reconhecimento de músicas que são tocadas ao seu redor. Ainda, o ACRCLOUD (ver Figura 8) é um serviço na nuvem, com um grande banco de dados musical. Além de oferecer o serviço de reconhecimento de músicas, ele também permite o reconhecimento de músicas pela web, como um teste do seu serviço.



**Tabela 5. Análise das soluções comerciais: Critério de Usabilidade**

CRITÉRIOS		Usabilidade				
		Visibilidade do status do sistema	Prevenção de erros	Flexibilidade e eficiência de uso	Estética e Design minimalista	Pouca interação homem/dispositivo
C	MusicID	sim	sim	sim	sim	sim
O	Shazam	sim	sim	sim	sim	sim
M	SoundHound	sim	sim	sim	sim	sim
E	Deezer	sim	sim	sim	sim	sim
R	Spotify	sim	sim	sim	sim	sim
C	SoundCloud	sim	sim	sim	sim	sim
I	Musixmatch	sim	sim	sim	sim	sim
A	ACRCloud	sim	sim	sim	sim	sim
L	Musipedia	sim	não	sim	não	sim

### 3.2.2. Soluções Acadêmicas

Em relação ao critério de Eficiência mostrado na Tabela 6, o comportamento das soluções em relação ao tempo foi encontrada documentação apenas para o Tunebot. A solução utiliza QBH e, segundo [Lab 2008b], o comprimento mediano de uma música cantada por um usuário é de cerca de 18 segundos de áudio, levando cerca de 5s para retornar resultados a partir do momento em que a aplicação recebe a música. Para comparação, a música mais longa recebida até o momento tem cerca de 48 segundos e a aplicação demora cerca de 13s para retornar uma resposta. O tempo de retorno é uma função de vários fatores, incluindo o tamanho do banco de dados e o comprimento da consulta, tanto em termos da duração total do áudio quanto do número de notas que o usuário cantou.

Cada solução possui ferramentas variadas para uso distinto no reconhecimento de músicas, o que gera diversas formas de utilização dos recursos. Por exemplo, o MusicMiner (ver Figura 14) possui ferramentas que utilizam os métodos de Classificação e Visualização através de Mapas Musicais Topográficos.

O AMUSE (ver Figura 10) e o Java MIR (ver Figura 12) utilizam o método de Classificação. Já o Tunebot (ver Figura 16 e 15) utiliza Query by Humming (QBH). O CLAM (ver Figura 11) utiliza Spectral Modeling Synthesis (SMS), e o MIRtoolbox (ver Figura 13) é um acoplado de ferramentas baseado em técnicas de mineração de dados. Por ser um pacote de ferramentas, são variadas técnicas para recuperação da informação musical que ele oferece, como por exemplo, clusterização e classificação.

Quanto ao critério de bitrate, das 6 soluções analisadas, não foi encontrada documentação que especificasse o bitrate utilizado para o fluxo de transferência das músicas.

Em relação ao critério de Adequação Funcional mostrado na Tabela 8, as soluções acadêmicas apresentam um modelo de desenvolvimento aberto, ou seja, seu código pode ser alterado por outros usuários, contribuindo para a resolução de problemas existentes nas aplicações. Das 6 soluções analisadas, todas possuem reconhecimento de músicas de forma aproximada (A), ou seja, permitem realizar buscas de músicas através de trechos de áudio e/ou voz e/ou codificação MIDI e/ou por conteúdo (metadados).

Todas as soluções permitem a inclusão de músicas criadas por usuários. Quanto à

**Tabela 6. Análise das soluções acadêmicas: Critério de Eficiência**

CRITÉRIOS		Eficiência		
		Comportamento em relação ao tempo	Utilização de recursos	Bitrate
A C A D Ê	<b>MusicMiner</b>	-	C e V	-
	<b>CLAM</b>	-	SMS	-
	<b>MIRtoolbox</b>	-	Diversas	-
	<b>AMUSE</b>	-	C	-
	<b>Java MIR</b>	-	C	-
	<b>Tunebot</b>	Até 13s	QBH	-

**Tabela 7. \***

Legenda: C - Classificação; V - Visualização; SMS - Spectral Modeling Synthesis; QBH - Query by Humming;

disponibilidade, com exceção do Tunebot, todas as soluções estão disponíveis para Linux, Mac OSX e Windows e seu acesso é off-line, ou seja, é necessário o download e instalação do projeto no dispositivo. O download e instalação das soluções é feito de forma gratuita. O Tunebot está disponível na Web e também possui uma versão beta para smartphones iOS, tendo o seu acesso on-line (Ver Figura 15).

O AMUSE e o Tunebot são as duas soluções que possuem integração com outras aplicações. O AMUSE faz parte do rol de ferramentas do MIRtoolbox. Já o Tunebot tem integração com o *Karaoke Callout*<sup>14</sup>, que ajuda a construir a base de conhecimento de músicas do Tunebot.

**Tabela 8. Análise das soluções acadêmicas: Critério de Adequação Funcional**

CRITÉRIOS		Adequação funcional						
		Disponibilidade	Modelo de desenv.	Integrações	Acessibilidade	Busca de dados	Inclusão de dados	Modelo de Pagamento
A C A D Ê M I C O	<b>MusicMiner</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>CLAM</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>MIRtoolbox</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>AMUSE</b>	-	A	MIRtoolbox	Off	A	sim	G
	<b>Java MIR</b>	Linux, Mac OSX, Windows	A	não	Off	A	sim	G
	<b>Tunebot</b>	Web, iOS	A	Karaoke Callout	On	A	sim	G

**Tabela 9. \***

Legenda: F - Fechado; A - Aberto; E - Busca Exata; A - Busca Aproximada; G - Gratuito; F - Freemium; P - Premium;

Como explicitado anteriormente, as soluções analisadas tem seu acesso off-line.

<sup>14</sup>*Karaoke Callout* é um jogo de karaokê para a plataforma iOS da Apple que permite aos usuários "tunebot"desafiarem uns aos outros para um duelo de canto [Shamma and Pardo 2006].

Diante disso, para o funcionamento correto de cada aplicação é necessário a instalação de projetos de terceiros. Desta forma, com a impossibilidade de funcionamento das aplicações, não foi possível analisar os critérios de usabilidade das soluções acadêmicas.

### **3.3. Discussão dos Resultados**

Conforme explicado na seção anterior, a análise de soluções foi dividida em dois grupos: soluções comerciais e soluções acadêmicas. Uma análise entre todas elas seria inviável, pois cada grupo se difere no motivo de uso e é voltado a públicos-alvo diferentes. As soluções comerciais são mais utilizadas por usuários finais. Já as soluções acadêmicas são voltadas a usuários pesquisadores da comunidade de MIR.

#### **3.3.1. Soluções Comerciais**

Com relação à análise das soluções comerciais, pode-se concluir que as soluções focam na usabilidade visando facilitar o uso para o seu público-alvo, que apenas as utilizam no seu dia-a-dia. Estes usuários não possuem muito interesse em como essas soluções funcionam, desejando apenas que a busca seja rápida e satisfaça os filtros informados.

Os testes realizados de reconhecimento e busca de músicas, por metadados ou por som, demonstraram que as aplicações atenderam bem a todas as funcionalidades oferecidas. Se o objetivo for o reconhecimento de músicas quando não se conhece nenhum valor de metadado, o Shazam e o Deezer são essenciais. Porém, se o objetivo for aquela música que não sai da cabeça e precisar cantar ou cantarolar, então o ideal é usar o SoundHound ou o MusicID. Se o conhecimento da letra da música for indispensável, é possível usar o Musixmatch.

Se o objetivo for playlists e download de músicas, para não ter a necessidade do uso de conexão com a Internet, o Spotify ou o Deezer atendem a demanda. Já o Musipedia, em particular, é uma solução voltada a usuários músicos, que compõem músicas através da codificação MIDI.

Em relação ao tempo, o ideal é a conexão de Internet Wi-Fi. Nestas condições, todas as soluções, exceto o Musipedia, fazem o reconhecimento/busca de músicas no tempo mínimo de 5s. Nesse caso, saber o objetivo do usuário com relação a desempenho é imprescindível para a escolha de qual solução usar.

Pode-se concluir que o método mais utilizado para o reconhecimento de músicas é o *Fingerprint*. Inclusive, o uso do serviço do ACRCLOUD tem sido incorporado em cada vez mais soluções para o reconhecimento das músicas, como é o caso do Musixmatch e do Deezer. Outro método bastante utilizado é o RPC, principalmente pelas soluções que permitem apenas buscas exatas, onde é feita a correspondência de texto para identificar arquivos de música digital, ou seja, através da informação do título, álbum ou gênero da música.

#### **3.3.2. Soluções Acadêmicas**

Com relação à análise realizada para as soluções acadêmicas, pode-se concluir que as soluções focam na otimização e desempenho no que diz respeito ao reconhecimento de

músicas. O desenvolvimento de tais aplicações é voltado para os usuários pesquisadores da comunidade de MIR, com o objetivo de trazer inovação na busca de dados musicais.

Todas as soluções incluem uma estratégia para busca por similaridade (a maioria delas utilizando técnicas de Classificação), são gratuitas e permitem a inclusão de dados musicais em seus repositórios. Ainda, a maioria são soluções off-line e estão disponíveis em diversas plataformas. Por outro lado, poucas soluções oferecem integração com outras soluções ou aplicações que tocam músicas. Inclusive apresentam um modelo de desenvolvimento aberto, ou seja, seu código pode ser alterado por outros usuários.

Houve dificuldade no uso das soluções acadêmicas e por este motivo, não houve a análise dos critérios de usabilidade, além de não ter sido possível o teste prático dos critérios de eficiência de desempenho, o que poderia ter enriquecido ainda mais a presente pesquisa. A documentação existente para cada proposta sugere que o desenvolvimento das soluções acadêmicas geralmente tem o objetivo de validar uma determinada pesquisa, sendo após oferecida gratuitamente à comunidade para futuras pesquisas e o aprimoramento da solução. Desta forma, a usabilidade nem sempre é o foco principal.

Por outro lado, certos processos já foram desenvolvidos e estão sendo utilizados como parte de uma nova solução, como é o caso do projeto AMUSE, que faz parte do rol de ferramentas do MIRtoolbox. Outro caso é o Tunebot, que possui uma versão beta desenvolvida para plataformas iOS, sendo um dos seus objetivos ajudar pesquisadores na área de reconhecimento de músicas que utilizam o algoritmo *Query by Humming*, facilitando uma pesquisa mais precisa do desempenho do mundo real do que seria possível com conjuntos de dados existentes.

Por fim, a escolha de um método para a recuperação da informação musical vai depender do objetivo da solução a ser desenvolvida. Neste sentido, quanto mais aprimorados forem os métodos e algoritmos de soluções acadêmicas, melhores soluções comerciais serão desenvolvidas para a utilização pelo usuário final.

#### **4. Considerações Finais**

Este trabalho teve como objetivo analisar soluções para a busca por similaridade de dados musicais, ou seja, soluções que não necessariamente buscam dados musicais apenas através do casamento direto de parâmetros de entrada para a busca, como título da música, palavras-chave ou um áudio como parte da música, mas também através do casamento aproximado (ou similar) destes parâmetros. Além de uma pesquisa aplicada sobre o que são dados musicais, sobre a forma como os dados são tratados e armazenados, para que possam posteriormente ser recuperados e ouvidos no dia-a-dia, este trabalho buscou reunir e comparar informações com o propósito de contribuir com futuros trabalhos que desejam desenvolver soluções para a busca por similaridade de dados musicais.

Pela observação das características analisadas nas Tabelas 1, 3 e 5, as soluções comerciais enfatizam a usabilidade, visando facilitar o uso para o público-alvo, que é o usuário final no seu dia-a-dia. O objetivo destas soluções pode ser o reconhecimento de músicas através de uma parte da música ou da voz, bem como a criação de playlists e download de músicas para uso off-line. O método mais utilizado para o reconhecimento de músicas é o Fingerprint.

As soluções acadêmicas analisadas nas Tabelas 6 e 8, por sua vez, enfatizam o

desempenho no reconhecimento de músicas. O desenvolvimento de tais soluções é destinado principalmente a usuários pesquisadores da comunidade de MIR com o objetivo de trazer inovação na maneira como são recuperados os dados musicais. Todas as propostas incluem uma estratégia para busca por similaridade (a maioria delas utilizando técnicas de Classificação), são gratuitas e permitem a inclusão de dados musicais em seus repositórios. Ainda, a maioria são soluções off-line e estão disponíveis em diversas plataformas. Por outro lado, poucas soluções oferecem integração com outras soluções ou aplicações que tocam músicas.

Como trabalhos futuros, sugere-se, primeiramente, a aquisição dos protótipos das soluções acadêmicas para que uma avaliação de usabilidade destas propostas possa ser realizada. Neste trabalho, devido a restrições de tempo, não foi possível ter acesso a essas soluções. Outra sugestão é um aprofundamento no estudo dos métodos e algoritmos utilizados para a recuperação da informação musical, como *FingerPrint*, Recuperação por Conteúdo e Classificação, bem como uma comparação para verificar os métodos mais eficientes. Por fim, o desenvolvimento e avaliação de soluções para recuperação de informação musical poderia contar com a participação de mais mulheres no programa de mentoria WIMIR<sup>15</sup>, incluindo soluções para busca por similaridade.

## Referências

- ACRCloud (2015). Acrcloud serviços de reconhecimento automático e confiável na nuvem acrcloud.
- Amatriain, X. (2004). *An Object-Oriented Metamodel for Digital Signal Processing with a focus on Audio and Music*. Phd thesis, universitat Pompeu Fabra, Barcelona.
- Amatriain, X. (2007). Clam: A framework for audio and music application development. *IEEE Softw* 24(1), page 82–85.
- Cartwright, M. and Pardo, B. (2012). Building a music search database using human computation. *Proceedings of the 9th Sound and Music Computing Conference (SMC 2012), Copenhagen, Denmark, July 12-14, 2012*.
- Daquino, F. (2012). A evolução do armazenamento de músicas [infográfico].
- Development, C. (2004). C++ library for audio and music - clam.
- Development, M. (2018?). Musipedia the open music encyclopedia.
- Dortmund, T. (2018?). Amuse - advanced music exploration.
- Downie, J. S. (2003). Music information retrieval (chapter 7). In *Annual Review of Information Science and Technology*, number 37, pages 295 – 340. Blaise Cronin, Medford, NJ: Information Today.
- F, S. (2012). itunes's new 1080p compares well to blu-ray quality, at a fraction of file size.
- Gomes, A. C. F. (2015). Representação da informação musical: uma análise a luz dos requisitos funcionais para dados de autoridade (frad). Graduação, Universidade de Brasília, Brasília.

---

<sup>15</sup><https://wimir.wordpress.com/mentoring-program/>

- Huq, A., Cartwright, M., and Pardo, B. (2010a). Crowdsourcing a real - world on - line query by humming system. *Proceedings of the 7th Sound and Music Computing Conference (SMC 2010), Barcelona, Spain, July 21-24, 2010.*
- Huq, A., Cartwright, M., and Pardo, B. (2010b). Crowdsourcing a real-world on-line query by humming system. In *Proceedings of the SMC 2010 - 7th Sound and Music Computing Conference*, Barcelona, Espanha.
- Lab, I. A. (2008a). Tunebot.
- Lab, I. A. (2008b). Tunebot dataset.
- Lartillot, O. (2013). Mirtoolbox user's guide 1.7.1.
- Lartillot, O. and Toivainen, P. (2007). Mir in matlab (ii): A toolbox for musical feature extraction from audio. In *Proc. 8th International Conference on Music Information Retrieval (ISMIR 2007)*, page 127–130, Vienna.
- McKay, C. (2010). *Automatic music classification with jMIR*. Phd thesis, CIRMMT and Marianopolis College, Québec.
- McKay, C. (2018?). jmir.
- McLane, A. (1996). Music as information. In *Annual Review of Information Science and Technology (ARIST)*, volume 31, pages 225 – 262. American Society for Information Science.
- Michels, U. (1992). *Atlas de música*. Madrid: Alianza, 2 edition.
- Morchen, F., Ultsch, A., Nocker, M., and Stamm, C. (2005). Databionic visualization of music collections according to perceptual distance. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, pages 396–403, London, UK.
- Muller, M. (2007). *Information Retrieval for Music and Motion*. ISBN 978-3-540-74047-6. Springer Berlin Heidelberg New York, Bonn, Germany.
- Nielsen, J. (1995). 10 usability heuristics for user interface design.
- of Jyväskylä, U. (2018?). Mirtoolbox.
- of Marburg, U. (2005). Musicminer - databionic musicminer.
- Pacheco, J. and Lopes, R. (2014). Descrição e procura de música. *Comunicação Áudio e Vídeo, Instituto Superior Técnico*.
- rer. nat. Christiane Gresse von Wangenheim, D. (2017). Critérios e recomendações ergonômicas - aula 3.
- SanDisk (2018?). Definição de bitrate.
- Santini, R. M. and de Souza, R. F. (2007). Recuperação da informação de música e a ciência da informação: Tendências e desafios de pesquisa. In *VIII ENANCIB – Encontro Nacional de Pesquisa em Ciência da Informação*, pages 1 – 14, Salvador, Bahia. Organização e Representação do Conhecimento Poster.
- Semidão, R. A. M. (2013). Dados, informação e conhecimento: elementos de análise conceitual. *DataGramaZero*, 14(4).

- Shamma, D. and Pardo, B. (2006). Karaoke callout: using social and collaborative cell phone networking for new entertainment modalities and data collection. In *Proceedings of ACM Multimedia Workshop on Audio and Music Computing for Multimedia (AMCMM 2006)*, Santa Barbara, CA, USA.
- Vatolkin, I., Theimer, W., and Botteck, M. (2010). Amuse (advanced music explorer) – a multitool framework for music data analysis. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, page 33–38, Utrecht, Netherlands.
- Wazlawick, R. S. (2012). *Engenharia de Software para Sistemas de Informação: Conceitos e práticas que fazem sentido*. UFSC, Florianópolis.
- Zuben, P. (2004). *Música e tecnologia: o som e seus novos instrumentos*, volume 1. Irmãos Vitale, second edition.